Technical Report 932

Automated Assembly Using Feature Localization

Steven J. Gordon

MIT Artificial Intelligence Laboratory

COVER

Automated Assembly Using Feature Localization

by

Steven Jeffrey Gordon

Submitted to the Department of Mechanical Engineering on December 20, 1986 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering.

Abstract

Automated assembly of mechanical devices is studied by researching methods of operating assembly equipment in a variable manner; that is, systems which may be configured to perform many different assembly operations are studied. The general parts assembly operation involves the removal of alignment errors within some tolerance and without damaging the parts. Two methods for eliminating alignment errors are discussed: *a priori suppression* and *measurement and removal.* Both methods are studied with the more novel *measurement and removal* technique being studied in greater detail. During the study of this technique, a fast and accurate six degree-of-freedom position sensor based on a light-stripe vision technique was developed. Specifications for the sensor were derived from an assembly-system error analysis. Studies on extracting accurate information from the sensor by optimally reducing redundant information, filtering quantization noise, and careful calibration procedures were performed.

Prototype assembly systems for both error elimination techniques were implemented and used to assemble several products. The assembly system based on the *a priori suppression* technique uses a number of mechanical assembly tools and software systems which extend the capabilities of industrial robots. The need for the tools was determined through an assembly task analysis of several consumer and automotive products. The assembly system based on the *measurement and removal* technique used the six degree-of-freedom position sensor to measure part misalignments. Robot commands for aligning the parts were automatically calculated based on the sensor data and executed.

Thesis Supervisor: Warren P. Seering
Title: Associate Professor of Mechanical Engineering

Acknowledgments

First I would like to thank my thesis advisor Warren Seering. Professor Seering puts his students first and was always willing to listen to my ideas and help direct my thoughts. He is perhaps the most energetic and optimistic person I have ever met and I have a great deal of respect for him. In addition to his time commitment he has taught me much about dealing with people and being a professional researcher.

I was fortunate enough to have selected an excellent thesis committee who took a strong interest in my problem. The members included Warren Seering, Dave Gossard, Tomas Lozano-Perez, and Dan Whitney. I believe their advice and comments were instrumental in making my thesis a worthy contribution. I would especially like to thank Dan Whitney for his thoughts on how the work should be organized.

I owe a lot to my family and friends who gave me support and the freedom to be somewhat reclusive for the past four or so years. My parents are responsible for shaping me into the person I am today; I will never be able to thank them enough.

My office mates and other AI lab members played an essential role in my research and social life during my stay at MIT. Al Ward and Neil Singer were part of the brainstorming team during the initial assembly cell design and are as much responsible for the basic concepts as I. In addition, Al designed the base plates and the interchangeable fingers and Neil designed the flexible pallets. Steve Eppinger helped me debug the cell and get it running. I would like to thank John Canny for his expert help with quaternions and feature detection and Steve Buckley for his authoritative AML advice. Thanks go to Mike Caine for his adjustable peg-in-hole.

Ken Pasch was always available to both bounce ideas off of and generate new ones. He is extremely sharp and almost always has good new ideas. I would also like to thank Beth, Dunbar, Karl, Eric, Mike, Rob, Marc, Terry and many others who gave me support.

Patrick Winston and the many others who make the AI lab a top notch research facility deserve recognition. This thesis describes research done at the Department of Mechanical Engineering and Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the System Development Foundation and in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124. Support for this research project is also provided in part by IBM.

Contents

List of Figures

List of Tables

Introduction

Chapter 1

1.1 Background and Motivation

The current capabilities of mechanical manipulators are inadequate to solve many industrial assembly problems. Although there has been some success with force and compliance controlled assembly machines (see [202] for an overview) and much success with passive compliant devices, notably the RCC or Remote Center of Compliance [104,133,204], most industrial systems depend upon precision assembly techniques (assembly is performed with no feedback other than precise positioning) to accomplish assembly operations. "Hard" automated assembly machines, machines specially designed for a single function, have been used very successfully, but they are capable of assembling only a single product and are inflexible to changes in product or part style. Because the flexible assembly problem is not well understood, the approach used in automating the problem has been to emulate humans by using some vision and force feedback. The approach taken in this thesis has been to solve the assembly problem by analyzing the task directly. Many of the techniques developed are fundamentally different from those used by humans.

1.1.1 Purpose of the Research

In order to develop technologically and economically viable flexible assembly systems the capabilities of present day systems must be extended so that they are able to handle a wide variety of part shapes, sizes, tolerances, and assembly operations without using excessively costly means. We address this need through theoretical and experimental investigations of programmable tools and sensors, and investigate feasibility through the development of a number of prototypes.

1.1.2 Flexible Assembly Systems and Batch Manufacturing

Flexibility is defined to be the ability to handle different parts and perform different assembly operations. It is one of the key issues in increasing productivity through automation in certain industries (see Section 2.1). Flexible assembly systems are electronically controlled mechanical systems which are capable of the production (primarily assembly) of morphologically different products. Theoretically, such a system is capable of producing sequential runs of many different products with relatively short setup times between runs. The size of the subset of products which a system can handle depends upon the size and weight of the elements as well as the operations necessary for the products' assembly.

Many of the process and mass production industries have been highly automated for some time. In contrast, processes used in batch manufacturing are almost entirely manual. The batch manufacturing system (a limited set of resources which is reconfigured to produce a number of different products) presents a number of unique problems to automation. Productive operation of the system requires sophisticated scheduling with almost all aspects of the system having some degree of flexibility.

1.1.3 The Rigid-Parts Assembly Process

An assembly task may be defined as follows

> Two or more parts are moved to a desired relative position within some tolerance. The process which juxtaposes the parts should not physically alter them unless it is a requirement of the task.

Thus, assembly is a positioning problem. According to Simunović [176]

> *The assembly process is strictly a positioning problem. Complete knowledge of the parts and ideal positioning devices would, at least in principle, make the assembly task a trivial matter. The imperfections of the real world are materialized as position errors in the physical assembly systems; these errors translate into an error in the relative position between the parts at mating;*

> *the resulting error in the relative positions between the parts at mating will cause interference between the geometry of the parts, and therefore not allow the parts to be assembled.*

In general, parts comprising an assembly are manufactured in batches of identical parts and are delivered to the assembly system in groups. In order to eliminate the relative positioning errors, the parts must be separated, grossly positioned, then mated.

The Three Phases of Rigid-Parts Assembly

Rigid-parts assembly may be broken down into three main phases independent of the type of system which performs the assembly.

1. Part acquisition
2. Part alignment
3. Part mating

The part acquisition phase entails part identification and gross orientation through conventional feeding mechanisms, machine vision or other sensing systems, or through manual techniques. The part alignment phase is typically performed by a manipulator and might also involve fixtures, sensors and search procedures. Part mating is the first phase where parts may touch one another. In this phase, either force or compliance control, passive compliance, or sufficient precision to allow non-contact mating is required. The performance of an assembly system in each of the latter phases is affected by the system performance in previous phases. This research attempts to increase the system performance in the part alignment phase and relax the requirements for the part mating phase.

Relationship Between Part Alignment and Part Mating

During the part mating phase, a direct position measurement of one part relative to the other is not generally available. Any necessary repositioning is driven by the forces generated between the mating parts. Techniques which have been used to eliminate positioning errors from force information include force and compliance control, logic branching, and passive compliance (see Reference [202] for an overview of these techniques). Passive compliance techniques are generally the simplest to implement for error correction during the mating phase. Relaxation of requirements for the passive compliance system is possible if the parts are aligned precisely enough during the part alignment phase.

Relaxation of the passive compliance requirements means that the assembly can successfully occur with a larger tolerance on the location of the center of rotational compliance and a larger tolerance on the magnitudes of the translational and rotational compliances. A successful assembly is one in which the parts are completely assembled without damage.

1.1.4 Methods for Eliminating Positioning Errors

Errors in part positions may be eliminated by one of two ways [176]

A Priori Suppression: Eliminate errors at their sources.

Measurement and Removal: Eliminate errors during or just before the mating process.

Hard automated assembly systems and some robotic assembly systems (such as the one described in Chapter 5) take the *a priori suppression* approach. These systems rely on accurate jigs and fixtures and precise actuator positioning. Humans and some advanced robotic assembly systems rely heavily on the *measurement and removal* approach to assembly. These systems use tactile (force) and visual (position) information to sufficiently align the parts. The sensor-driven assembly system described in Chapter 6 takes this approach.

A Priori Error Suppression

Elimination of errors by the *a priori suppression* method entails controlling error propagation. In general, when components are originally manufactured, their position is well known. For example, the location of features of parts machined in a milling machine are referenced to the cutting tool up until the point where the vice is released. If the location of a part were precisely maintained from the point of manufacture to the point of assembly, only small errors would need to be eliminated for the mating phase. The *a priori suppression* method also includes systems which do not constrain the location of parts from their point of manufacture, but which orient parts at some later time then fixture them in pallets or jigs prior to their assembly.

Sources of position errors include

- Loss of location information after machining or molding operations.
- Loss of location information after finishing operations such as plating, polishing, tumbling and cleaning.
- Finite precision orienting techniques (e.g. bowl feeding).
- Finite precision pallet, assembly jig, or gripper fixturing.
- Finite precision manipulator positioning.
- Significant part tolerances.

In order for the *a priori suppression* method to be successful, errors from all pertinent sources must be controlled.

Error Measurement and Removal

In the *measurement and removal* method for eliminating errors, the relative position of mating part features are measured either directly from position measurements or indirectly from force measurements. After the measurements are made, the manipulator reorients the parts nullifying measured misalignments. In an ideal system using this approach, sensor accuracy and manipulator motion resolution are precise enough to mate the parts without interference between surfaces. A more practical system would rely on the forces generated between mating surfaces to correct any remaining errors with the aid of the passive compliance of the system.

1.1.5 Comparison of the Two Methods of Error Removal

Advantages and Disadvantages of Each Method

A system which uses the *measurement and removal* method of eliminating position errors is more flexible than a system based on the *a priori suppression* method; that is, less specialized tooling is required to perform a large variety of tasks. The system which uses the *measurement and removal* method is relatively insensitive to the accuracy and wear of jigs, grippers, and pallets which locate the parts and the assembly. Since measurements of the mating features are made, the system is also relatively insensitive to large non-mating feature part tolerances and imperfections (such as burrs). The jigs which locate the parts need to locate them to within the acceptable range of the sensor. This may be as large as an order of magnitude bigger than the range allowable in the *a priori suppression* method (see Section 6.6.1). In some cases it may be desirable for the part jigs (including grippers and pallets) to only firmly hold the parts and not accurately locate them. Thus the *measurement and removal* method supports the use of universal grippers and pallets.

Using the measuring method allows assembly operations to be performed more reliably. Because the part features are sensed, there is additional assurance that the part is in the proper position prior to mating. It may also be possible to more accurately align parts. In addition, successful assembly operations may be performed without calibrating the robot to all of the pallets and jigs in the cell. The system operates somewhat uncoupled from the absolute positioning of the manipulator; that is, all commanded motions are relative to sensed positions with respect to the world frame. Because of this, offline programming of assembly tasks is easier and requires less absolute position references and online teaching and calibration.

There are, however, a number of disadvantages to the *measurement and removal* method. First, a finite time is required to perform the sensing. Depending on the type of sensor and the stage of the assembly process in which the measurement is made, the assembly procedure might be slowed. For a vision sensor with the camera mounted offboard of the manipulator, the system must allow sufficient time for the manipulator to come to rest (let all vibrations settle). Time is also required to grab a frame ($\frac{1}{60}$ sec.) and process the image. The system also has finite measurement accuracy which might not be appropriate for all tasks. A vision based system is also sensitive to the surface reflectance properties of objects and surface orientations with respect to the sensor.

A limiting consideration in using a vision based *measurement and removal* method is that not all features are easily sensed. It is likely that features comprised of relatively simple primitives (e.g. quadric surfaces) may be sensed with a technique similar to the one presented in Chapter 4, but more complicated features may create problems. One study has showed that about 85 percent of "all manufactured parts" may be accurately modeled by planar, cylindrical, and spherical patches [81]. Although the accuracy of this estimate is questionable, the estimate is at least promising. Even if features of any shape could be located with the sensor, it is likely that the system would not be able to locate features with large size differences. It might, however, be possible to design a system with an adjustable field of view to accurately locate different sized features.

Which Method is Better?

Both the *a priori error suppression* and the *measurement and removal* methods of error elimination may be successfully used for robotic assembly tasks. The best method to use depends upon the requirements of the task. The *a priori error suppression* method is relatively insensitive to the shape of the parts being mated; however, it is relatively inflexible. A particular set of hardware must be used to accurately fixture to parts. The *measurement and removal* is highly flexible since almost no specialized mechanical fixtures are used, but the types of part shapes which may be sensed and assembled is limited.

With further development of the sensing system proposed in this thesis (or other sensing techniques), the sensor-driven assembly technique will be capable of handling a larger number of part shapes. As compared to the *a priori suppression* method it will be much more flexible and cost effective for industrial assembly systems operating in a batch production mode.

1.2 Overview of the Thesis

This thesis discusses issues in programmable mechanical assembly systems. In this chapter we have introduces the problem and suggested approaches to its solution. Two methods were proposed to eliminate the errors in part alignment: *a priori elimination*, and *measurement and removal.*

Chapter 2 gives the background on the assembly problem and an elaboration of the two solution techniques. A literature review of research in programmable assembly systems is followed by a survey of operations involved in certain mechanical assemblies. Assembly operations are classified with respect to the magnitude of difficulty for a single chain manipulator. A review of failure modes for the canonical peg-in-hole assembly operation is presented. Errors in robotic assembly systems are analyzed and applied to both the *a priori suppression* and the *measurement and removal* methods. Accuracy specifications for a part position sensor are calculated.

Requirements for an industrial sensing system for measuring part alignment errors are discussed in Chapter 3. A literature review of ranging techniques is given and two techniques are analytically explored. Arguments for using a light-stripe vision system for the part-position-sensing function are presented.

Chapter 4 discusses the details of the development of a light-stripe based part position sensor. First literature dealing with research in extracting accurate information from noisy images is reviewed. Both optical and geometric fundamentals of the light-stripe technique are then presented. Methods for extracting and quantifying the accuracy of information from light-stripe images are then explored. These include an error analysis of finding straight line features corrupted by quantization noise, methods for determining the accuracy in finding the center of a light-stripe using three different techniques, and a technique to combine redundant information from multiple light plane illumination of a part feature. The hardware and algorithms necessary to extract six degree-of-freedom measurements from a single light-stripe image are also presented.

Chapters 5 and 6 describe prototype assembly systems using the two methods of error elimination presented in this chapter. The system in Chapter 5 uses a number of flexibly designed tools and an industrial robot to control the propagation of position errors during the assembly process. The assembly system in Chapter 6 is vision sensor based. The accuracy of a prototype light-stripe vision system used for part position measurements is investigated in a specially constructed test bed. Details of the sensor calibration are also given.

Chapter 7 contains a discussion of what additional research is necessary to develop the prototype systems into industrial systems which may be used in manufacturing facilities. Uses for the sensing technology in addition to part position sensing for assembly operations are also presented. Finally conclusions and technical contributions of the work are enumerated.

Programmable Assembly Systems

Chapter 2

2.1 Literature Review of Research in Programmable Assembly

Worker productivity has increased steadily throughout recent history. Within the last decade, application of automated manufacturing technology has resulted in dramatic changes in rates of productivity growth within those industrialized nations choosing to invest the necessary capital and human resources. Tesar rated a number of elements which produce increased productivity [192]. He found that technology produces 38.1 percent, capital - 25.4 percent, labor quality - 14.3 percent, economics of scale - 12.7 percent and resource allocation - 9.5 percent. He also notes that all western trading partners had higher productivity growths than the United States.

The Labor Force

The work force in the United States has recently undergone a major shift in worker qualifications which could result in a growing demand for manually skilled labor. According to Merchant [126], 60 percent of the current U.S. workforce hold degrees from a secondary school, while 50 percent of those entering the workforce have a college or university education. According to Catalano [42] there will be a shortage of manual labor by the year 1990 assuming a moderate GNP growth and limited productivity increases from automation. Merchant and Catalano suggest that since there are fewer people who will be entering into the manual labor force, manufacturing industries must either automate a number of their operations or drastically improve the efficiency of the operations. Catalano gives estimates of manual labor shortages in the US, but these are probably not accurate since an increased foreign labor market is not taken into account and the sources of the GNP will shift other industries.

Manufacturing's Economic Contribution

Merchant notes that although manufacturing industries account for 33 percent of the international gross national product (24 percent of the U.S. GNP) while service industries make up 50 percent (63 percent in U.S.), it produces 66 percent of the wealth (65 percent in the U.S.). Many of the process and mass production manufacturing industries have been highly automated for some time. In contrast, processes used in batch manufacturing are almost entirely manual. The batch manufacturing environment presents a number of unique problems to automation. A single system (a set of tools and resources) is frequently reconfigured to produce a number of different products. Productive operation of the system requires sophisticated scheduling with almost all aspects of the system having some degree of flexibility. Anderson [10] notes that since 75 percent (by value) of all U.S. discreet engineered products are produced in the batch mode, national productivity can be significantly increased by making the batch manufacturing process more efficient.

Automation in Manufacturing

One method of improving batch production efficiency is to automate with computers and computer controlled machinery. Anderson [10] demonstrates that machine shops have reduced both direct labor cost and manufacturing time by 75 percent with the use of numerically controlled and computer numerically controlled machines. Increased computerization in the batch production industry will likely increase the utilization of capital equipment as well as increase the quality and uniformity of the products produced. Presently, parts being processed in the job shop environment are idle about 90 percent of the time [10]; whereas, in an automated job shop environment, in-process inventory is substantially reduced. In addition to the economic benefits of automation, the worker will be relieved from boring trivial tasks which may be readily accomplished by machines and is more likely to have more interesting skilled work. Yonemoto of the Japanese Industrial Robot Association (JIRA) [207,208] shows that some improvements associated with robots are increased productivity, humanization of working life, increased labor safety, improved product quality and early return on investment. Many US managers would probably disagree with Yonemoto's last "improvement" since low return on investment has been one of the major factors retarding factory automation through robot installations.

Economics of Assembly

Anderson estimates that assembly accounts for about 35 percent of the production cost for discreetly engineered products. Nevins and Whitney of the Charles Stark Draper Labs [132,133] have studied the science of assembly and have classified the 3 modes of assembly. Manual assembly is appropriate for products with low production volume. Low fixed costs are also associated with this mode so there is no economy of scale. The manual assembler has the characteristics of being very flexible and easy to train. He has excellent sensory capabilities, but may tend to lack reproducibility and get bored. Assembly via fixed automation is appropriate for products with high volume constraints. Fixed automation typically has high fixed costs and high efficiencies. These systems are not very flexible and tend to fail due to part jams while there is usually little sensory capability. Programmable automatic assembly has medium fixed costs and is appropriate for medium production volumes. It has medium efficiency and is capable of responding to sensory inputs and learning new tasks. Nevins and Whitney have also studied the amount which is invested in assembly in a number of different industries. Motor vehicle and radio and television industries have about 30 percent of direct labor attributable to assembly. They note that a better indication of savings which comes from automating the assembly process is percent value of shipments due to assembler's pay. These figures are 4.7 percent for the motor vehicle industry and 3.8 percent for the radio and TV industries. Nevins and Whitney fail to take into account additional savings from automation due to increased organization, lower in-process inventory, lower personnel and paper work overhead costs, and higher efficiency. Boothroyd [33] has also studied the amount of labor and manufacturing costs attributable to assembly based on a 1967 census of manufacturers. He found that motor vehicle and telephone industries have about 50 percent of all production workers involved with assembly. Other industries such as motorcycle, aircraft, farm machinery, and refrigerator and freezer have from 20 to 40 percent of labor involved in assembly. Boothroyd postulates that assembly accounts for about 50 percent of the total manufacturing cost for a product. The apparent large discrepancies between costs estimates of Boothroyd, Anderson, and Nevins and Whitney are most likely due to inaccuracies in estimation, comparison of just wages to all costs and other factors previously stated.

The Assembly Process

The most frequent assembly operations and part orientations during assembly were studied by Nevins and Whitney by examining ten products. They found that 33 percent of the assembly operations are peg in hole insertions, 27 percent are screw insertions and 12 percent are push and twist operations. Most other operations include multiple peg in hole, force fits, insert peg and retainer (all less than 10 percent), flip part, provide temporary support, remove temporary support, remove locating pin, weld or solder, and crimp sheet metal (all less than 3 percent). Most of the operations were unidirectional (e.g. 80 percent of all peg in hole insertions were from the same direction). Nevins and Whitney also cite surveys from General Motors and John Deere which deal with the average mass of a part which is handled during vehicle assembly. General Motors found that 90 percent of the parts in an average automobile are less than 2 kilograms (4.4 pounds) while John Deere reported that 80 percent of the parts in their farm equipment weighed less than 4 kilograms (8.8 pounds).

The Canonical Assembly Operation: Peg-in-Hole

Since it was found that peg in hole insertion dominated assembly tasks, researchers at the Draper Labs extensively researched the subject [104,132,133] They studied clearance ratios (clearance/diameter) of close fits and found that similar types of parts had similar clearance ratios. Bearings had the smallest clearance ratios of the parts which were considered. Contact forces were analyzed and criterion for wedging and jamming of parts were formulated. The forces during the three stages of insertion namely chamfer crossing, one point contact and two point contact could be calculated as a function of offset of centers and insertion depth. Whitney and Nevins made a major breakthrough in the science of assembly with the development of the remote center of compliance (RCC) [204]. This is a passive device which is capable of providing a large degree of translational and rotational compliance in directions orthogonal to the direction of insertion while remaining stiff in the direction of insertion. In addition, the device locates the system's center of compliance at the bottom center of the peg being inserted. Thus the RCC is capable of apparently "pulling" the peg into the hole from the bottom. An instrumented RCC has been developed which can be used as a teaching aid for a robot by automatically finding the exact location of a hole or as a sensor for an active control system for the robot. Whitney and Nevins have also done a number of studies on chamferless and compliant part insertion. Takeyasu, Goto and Inoyama [187] report on the Hitachi Hi-T-Hand which is also able to do close tolerance peg in hole insertions using active feedback. However, this manipulator performs the task somewhat slower than the RCC.

Design and Classification of Robots

Because of their intrinsic flexibility, robots are often envisioned or utilized in programmable automatic assembly stations. The literature contains a large range of opinions as to the optimal design for an assembly robot [66,131,159,187,194]. Since Japan possessed 69 percent of the industrial robots in operation in 1979 while the United States possessed only 16 percent (using similar definitions of robots) [66], many of the studies on robots were done in Japan. McPherson [124]] discusses the history of robots in Japan as well as some current data on robots. He reports on JIRA's survey on reasons for the introduction of robots which showed labor savings as being the most frequenly given response (44.5 percent). Other responses included improvement of working conditions (24.9 percent), versatility of production systems (13.5 percent), facilitation of management (8 percent), and 9.1 percent due to other reasons. JIRA also predicts that assembly robots will move from 10 percent of the robot market in 1980 to 17 percent in 1985 and 22 percent in 1990. JIRA's identifies 6 classifications of robots:

Manual Manipulator - A machine directly operated by a human.

Fixed Sequence - A machine which may be programmed for a particular task but whose reprogramming ability is minimal.

Variable Sequence - Same as the fixed sequence robot but the machine's program is easily changed.

Playback Robot - This machine is only able to memorize sequences directly taught by a human.

Numerical Control - A machine which performs according to digital information on sequence, position, etc.

Intelligent Robot - This machine uses vision, sensors, etc. to determine position, action, rate, etc.

Gevarter reviewed a number of other Japanese studies of robots in his report to NBS and NASA [66]. A 1981 survey of the uses of robots in Japan showed unloading and loading the most frequent at 40 percent. 21 percent of the robots in Japan were used for transfer and sorting, 9 percent for palletizing, 6 percent for welding, 4 percent for work maintenance, 3 percent for assembly, 2 percent for spraying, and other uses such as pouring, screwing, and riveting comprised 15 percent of the robots. The distribution of types of robots produced in Japan are: manual manipulators - 10 percent, fixed sequence - 67 percent, variable sequence - 7 percent, playback robots - 10 percent, NC robots - 5 percent and intelligent - 1 percent. The results of a 1980 JIRA users survey of necessary research areas (in order of preference) is:

1. More degrees of freedom
2. More compact robots
3. Higher speed robots
4. A larger assortment of attachments
5. Easier reprogramming
6. Greater reliability
7. Increased working volume
8. Increased payload
9. Increased accuracy
10. Tactile sensing
11. Vision
12. Pattern recognition
13. Increased memory
14. Higher mobility
15. Coordinated control of multiple robots

A 1980 JIRA survey of current research areas of Japanese robot manufacturers shows .1 percent involved with increasing robot speed, .7 percent involved with making robots more compact, 8.6 percent with computer control, 8 percent with lighter weight robots, .7 percent with modular interchangeability, 5 percent with object recognition, 4.6 percent with increased payload, 3.8 percent with improved actuators, 3.8 percent with self diagnosis, and 3.8 involved with adaptive control. Still another JIRA study as reported by Yonemoto [207] of the distribution of robot sales to different industries reveals electric machine industry - 36 percent, automobile industry - 30 percent, plastics molding - 10 percent, metal working industries - 5 percent, and metal working machinery industries - 4 percent.

Seering [168] contends that robots are presently designed to emulate humans and not designed to perform important assembly tasks. Mechanical manipulators should not be constrained to move at human speeds, carry human compatable payloads, work with the same precision, and have the same sensing capabilities of humans. This philosophy was adhered to in many aspects of this thesis.

Costs in Assembly Systems

In order for programmable assembly systems to be implemented in the industrial environment, they must prove to be more economic than conventional modes of assembly. According to Elbracht and Schaler [57], the economics of programmable assembly may be compared to the economics of manual assembly by comparing the costs of necessary capital equipment versus the cost of labor. They note that an acceptable cost for automated equipment depends on both the number of shifts being considered as well as the country where the installation will occur. A number of authors discuss methods for predicting the economic feasibility of flexible assembly systems. Boothroyd [31,33] and Dewhurst and Boothroyd [50] calculate per part costs versus volume per shift-year for a number of assembly systems. Systems which were considered are totally manual systems, manual systems with feeders, indexing type fixed automatic machines (all workpieces indexed simultaneously), free transfer machines (buffers between each workhead), programmable workheads (robots), two arm programmable stations, and a universal assembly station consisting of two or more arms with programmable end effectors and programmable feeders. He found that manual types of assembly are economic below about 35,000 units/shift-year, fixed automation assembly is economic above about 850,000 units/shift-year, and programmable assembly is economic between these production volumes. Boothroyd demonstrates that the volumes where the various modes of assembly become economic vary strongly as a function of number of product design changes per year, number of product styles to be produced, and number of different products to be produced. Nevins and Whitney predicted similar levels of production volume where the various modes of assembly become economic using somewhat different models [132,133]. Their models for cost per assembly (finished unit) were based on payback period methods and are as follows:

Manual:

$$Cost/Assy = AssyTime/Part \times LaborCost \times Parts/Assy$$

Fixed automation:

$$Cost/Assy = \frac{Parts/Assy \times MachCost/Part}{PaybackPeriod \times Volume}$$

Programmable assembly:

$$Cost/Assy = \frac{Parts/Assy}{PaybackPeriod} \times \left(\frac{StationPrice \times Time/Part}{Seconds/Year} + \frac{ToolingCost/Part}{Volume}\right)$$

Depending on the payback period and labor costs, the results of analysis on 10 part units are: manual assembly is economic up to 100,000 units, programmable assembly is economic from 100,000 units to 2 million units and fixed automation is economic above 2 million units.

Benedetti discusses another method of calculating the most economic mode of assembly by optimizing a profitability condition with respect to some volume of production [21]. This condition compares the costs involved with the purely manual operations to the costs of automated machine operation, automation machinery capital costs and the costs of manual intervention. Benedetti notes that this method is not based on discounted cash flow techniques and performs other analyses which are. From these models, he calculates the best method of assembly based on both cycle time and annual production volume assuming some rate of return and some utilization period. He also determines the amount of investments available for automation as a function of operator reduction.

To accurately predict the actual costs involved with the implementation of a flexible manufacturing system, any model used should be based on discounted cash flow techniques. The Boothroyd, and Nevins and Whitney models could be made more precise by taking into account the time value of money. None of the above economic models take into account the loss of business if automation is not pursued. In other words, companies which do decide to automate may attract a larger market share due to their increased flexibility and shorter delivery time. Companies which do not automate may find their share of business taken over by a more productive company using flexible automation. Another concern when considering the economic feasibility for a newly automated system as cited by Elbracht and Schaler [57] is the fact that automated equipment costs are presently rising at a slower rate than labor costs.

Design for Assembly

Assembly research involving the programmable automatic assembly of parts which were designed to be manually assembled without redesign have shown that system implementation is both uneconomic and difficult [84,119,132,181]. These findings imply that design or redesign of parts for programmable assembly is extremely important if a newly designed system is to succeed economically. Boothroyd has documented methods to improve designs for ease of assembly [31,33]. He suggests:

1. Reduce number of parts.
2. Unidirectional assembly.
3. Chamfer insertion interfaces.
4. Make parts locatable.
5. Use a base part.
6. Layered assembly.
7. Simple fastening operations.

Other authors [51,109,123] suggest additional methods for improving designs for assembly such as good interfacing between base part and fixtures, logical assembly order, designing for facilitation of inspection, keeping tight part tolerances or using a passive remote center of compliance, designing parts with a low center of gravity for stability, protecting fragile surfaces, providing a suitable gripper and feed track surface, and avoiding or not using separate fasteners.

Boothroyd et al. [32,33] and others [11,51,109] also suggest improvements of designs to facilitate automatic feeding including designs which decrease the likelihood of part tangling and hooking, maximize part symmetry or exaggerate asymmetry, have smooth surface finishes for feeding, use special orienting faces, use high quality components, have part geometries which fit into magazines, and designs which use preoriented parts on tapes. Lewis also suggests a clean assembly environment to avoid feeder jams [109].

A number of authors make a point of looking at each part and making sure that it is necessary in the total product [11,31,33]. Boothroyd rates the efficiency of a given design with respect to the minimum number of parts and minimum handling and assembly time. Design efficiency is defined as:

$$\eta = \frac{Theor.Min\#Parts \times (Nom.HandleTime + Nom.AssembleTime)}{ActualAssembleTime}$$

Boothroyd has designed a system which will help the designer increase the efficiency of a design and predict the costs and the amount of time necessary for assembly. The system classifies each part in an assembly with two digits the first of which quantifies the amount of symmetry and ease of grasp. The second digit is based on the mode of insertion or fastening. The technique was designed with manual assembly in mind but may also be used for programmable automated assembly.

Andreasen [11] classifies the different types of assembly structures as being frame, staked (some components hold others), composite product (different materials), base component (base for transport and assembly), modules, and building block. Djupmark [51] rates a number of fastening techniques with respect to ease of implementation in an automatic assembly workhead (from simplest to most difficult):

1. Pressing
2. Snap joints
3. Lap joints
4. Baking in
5. Welding
6. Riveting
7. Screws
8. Pins and Rings
9. Crimping
10. Soldering
11. Gluing

The above list is machine dependent and is most likely not accurate for new machine designs.

Techniques which may reduce the cost of programmable assembly automation as cited by Redford [159] include increasing the speed of robots (although it is likely that less than an order of magnitude increase is possible), use of limited capability and inexpensive robots, versatile, inexpensive grippers (using more than one gripper on a robot arm, designing a programmable gripper, designing parts to minimize gripper change or assembling a number of assemblies at a time to minimize gripper changes), identification of assembly families, and lower feeding costs.

Parts Feeding

Redford analyzes a number of different types of feeders with respect to cost of implementation in a programmable assembly center [159,160]. The types of feeders considered were:

1. Dedicated (bowl feeders).
2. Multi-part (5 parts, 1 drive, different orienting tracks).
3. Programmable.
4. Dedicated feeders serving more than 1 robot.
5. Feeders with vision.
6. Magazine systems (better utilization of feeders, secondary inspection before assy).

   (a). Manually Loaded. (b). dedicated feeders.
   (c). Multi-part feeders. (d). programmable feeders.
   (e). Loaded by prior manufacturing operation.
7. Manually loaded feed tracks.

Feeding costs depend on:

1. Material handling cost. 2. System tending cost.
3. Fault correction cost. 4. Change over cost.
5. Equipment depreciation cost. 6. Tooling depreciation cost.

The results of Redford's feeder cost analysis were based on a study of two product families with volumes of 200,000 units per year. One family consisted of 66 product types and the other consisted of 20 part types. Variable batch sizes were considered from 50 to 4350 units. Results showed that all systems except magazines loaded from prior manufacturing operations exhibited dramatic cost increases at batch sizes below 450 units. All cost versus batch size curves were parallel at higher volumes (independent of batch size). Multi-part feeders, feeders with vision, and magazines loaded at prior manufacturing operations were the most economic feeding techniques. Manual loading of robots was more economic than programmable or dedicated feeding. It was suggested that a mix of different feeder types is probably best.

Conventional bowl feeders were examined by Boothroyd [32,33] who determined the variables responsible for feed rate. Feed rate is a function of orienting efficiency (dependent upon the number of natural resting states of a part) and track conveying velocity. Conveying velocity is a function of ramp angle, vibration angle, frequency of vibration, coefficient of friction, and load sensitivity (change of part velocity due to the amount of material in the bowl). A number of non-conventional feeders are also discussed by Boothroyd such as non-vibratory feeders for parts with simple geometries such as discs and cylinders. Out of phase feeders are also described whose main attribute is an increase in feeding efficiency due to a decoupling of the two principal directions of vibration. Boothroyd is also involved with the design of belt feeders some of which are programmable with simple optical sensing capability. In addition, he has derived a part feeding and orienting coding system which can help designers to design parts for ease of feeding. The code is based on the shape of the part (relative dimensions), the amount of part symmetry, and protruding or other orientable features. The system points out difficult to feed parts as well as serving as a guide for the designer.

A number of researchers have developed "smart" feeders to increase the flexibility of automated assembly systems. Hill and Sword [89] use vision to check part orientation. If reorientation is necessary, parts are turned over by being pushed off a ledge and rotated on a rotary table. The cycle time including visual processing time is 15 seconds. Suzuki and Kohno of Hitachi [185] report the use of a multi-level bowl feeder with no orientation tracks which uses adjustable wipers and dish-outs to partially orient parts. After being partially oriented, the part is pressed up against datum planes and visually scanned. The part's orientation is then determined and either the feeder flips it into the proper position or the robot reorients it before insertion in the assembly. This type of feeder is very flexible and can accommodate a wide variety of part shapes and sizes. The Swedish Institute for Production Engineering uses a simple linear array camera with a belt conveyer equipped with wiper blades to flexibly feed parts [51]. Another flexible feeder which uses vision is reported by Heginbotham [85]. The system consists of a bowl feeder feeding onto a belt. From the belt, the part is pushed past fiberoptics along two perpendicular walls. The fiberoptics terminate at a linear scan camera which send the visual data to a computer. The robot which reorients the part before it is assembled is also capable of rejecting parts which do not pass the visual inspection.

Selective Assembly

A system with fairly sophisticated inspection facilities is discussed by Camera and Migliardi [39]. It uses the DEA PRAGMA robot and automatic inspection equipment to dimensionally and functionally inspect precision parts such as automobile crankshafts or injectors. They note that instrumented grippers may also be used for some gross dimensional inspection. After being inspected, the parts are placed in tolerance groups for later insertion into other parts belonging to appropriately matched groups.

Time Motion Studies

A major concern when considering the costs of a manual system versus a robotic system is the reduction of thoughput time. Rogers of Unimation demonstrates how robot time and motion studies may be used to compare robotic manufacturing with other modes of manufacturing [161]. These studies are similar to manual Methods-Time Measurement studies (MTM) and may be used to help find optimal manufacturing sequences, balance a prodution line and compare robots from different manufacturers on an individual task basis. He notes, as do others [21,33,50,57,66,85,112,137,138,139,140,141,148,159], that asembly costs are very sensitive to the speed of a robot (much more than to the price of the robot). Since robots are very consistent, robot time motion standard times can be much more accurate than MTM standard times. Rogers discusses three techniques for determining a standard time for robot tasks. The first and simplest technique is just to calculate the average time needed to perform a number of typical tasks with a typical number of tool changes. The second technique takes into account the type of task being performed and the third technique takes the robot control scheme into account (accounting for ramped and different maximum velocities). Nof et al [137,138,139,140,141,148] take a similar approach to arrive at standard times for robot tasks. They compare Robot Time Motion (RTM) times directly to MTM times for a number of tasks. They find that humans are not capable of performing all of the tasks that robots can perform. Of course the converse is also true. A comparison of times necessary for the assembly of a fuel pump for both manual and robotic systems demonstrated that the human was capable of doing the task about 8 times faster than the robot. Other studies such as the Draper Lab alternator assembly have yielded similar assembly time ratios [132,133].

Integrated Factory Control

Anderson has suggested that maximum impact of computers on manufacturing systems will be the complete, real-time computer cognizance and control of all processes and resources allowing precise scheduling and allocation [10]. He also states that a system like IBM's COPICS system (Communications Oriented Production Information and Control System) which is a factory data collection system is necessary for such computer control. Fisher et al [62] cite authors who attempt to model the facility and planning of a computer controlled manufacturing system using a number of different techniques many of which are based on closed queuing networks. A number of simulation models are also cited. Fisher et al. construct a model based on probabilistic analysis of part recirculation where a part is circulated through inspection and rework until it is within specification. Gershwin [65] and Kimemia and Gershwin [101] consider the control of a computerized manufacturing system with buffer storages between each work station. They calculate line efficiencies based on mean time between failure and mean time to repair for each machine. The analysis, which is based on optimal stochastic control models, becomes extremely complex for more than 2 or 3 work stations and thus has practical limitations.

Schröder [167] discusses how machines and humans may be optimally integrated in an assembly environment. System configurations are discussed which make manual assembly independent of machine cycle time. It is suggested that lines be put together in a modular fashion so that they may be easily changed. Grouping manual stations close to one another yet separate from automatic stations will promote worker communication and avoid worker dislocation. He gives configurations for grouping manual stations together even when automatic operations are interspersed. Schröder notes that if manual and automatic operations are not mixed in this manner, even work distribution is not always possible.

Worker Acceptance of Automation

The implementation and employee acceptance of a computer controlled planning system is presented in a case study by Shaiken [170]. TOPS (Total Operation and Planning System) was implemented in a large tool and die shop of a large automobile manufacturer. Its function was to control the complex scheduling of operations involved with producing a die. The system was despised by the workers after it was implemented. They thought that the system took the skill out of their job and that the time they spent working was constantly being monitored. Part of the problem with the computerized planning system was that it tried to quantify a highly skilled job which takes over a decade for a good toolmaker to master. The computer was not able to make "seat of the pants" type decisions which are sometimes essential in die manufacture.

Although labor unions are concerned with the short term consequences of increased automation, they know that robots will benefit society by increasing productivity and relieving people from dangerous and undesirable jobs [83]. According to the UAW, after a new piece of automation has been installed, it is still important that the worker have a sense of security and obligation.

Programmable Assembly System Implementations

To date, just more than a handful of flexible manufacturing systems are in operation with only a few involving assembly. Many of the systems are experimental with the exception of some Japanese systems which are involved in a significant part of the manufacturing process. Over 5 years ago Westinghouse undertook its Adaptable Programmable Assembly System (APAS) project with the intent of developing a state of the art system [1,119,181]. One of the firsts tasks Westinghouse studied was choice of a product line which was suitable for programmable assembly. Abraham [1] points out that over 60 product lines were considered and after an intricate process of elimination, 3 were chosen as possible APAS candidates. Each line was rated on (In order of importance):

1. Use of APAS technology
2. Degree of transfer
3. Social desirability
4. Inspection and recognition
5. Fixturing tooling
6. Economics
7. Product redesign

Four system configurations were considered and evaluated separately for each product line. The first configuration involved separate subassembly and final assembly stations. Another configuration used a single arm robot with off-line parts feeding. The third system utilized lower degree of freedom arms in a line with off-line parts feeding and the fourth configuration used many assembly stations with off line parts feeding. It was found that the system configuration utilizing lower degree of freedom manipulators is best for short cycle time products with limited style variations. The most significant costs were equipment and cycle time with cycle time being the most significant. The use of 1 to 2 second cycle times was suggested for cost reduction. From the initial study, fractional horsepower electric motors were chosen for system implementation. To date, the system has been implemented for the end bell assembly of the motors. The system incorporates a number of fairly new technologies such as programmable belt feeders, multi-part handling end effectors, and visual inspection. Although one of the original goals of the system was that it should be capable of assembling existing parts, it was found that these resrictions on part redesign made implementation extremely difficult and expensive.

Nippondenso Corporation in Japan uses flexible assembly lines for manufacturing automobile instruments [131]. They have developed their own simple non-sensing robots to do the assembly because fast, inexpensive, limited degree of freedom robots were unavailable. They made the following evaluations of humans and a number of robots:

TECHNIQUE	CYCLE TIME (SEC)	TEACH TIME (MIN)
Humans	1.4	1
Nippondenso robot	1.9	100
SCARA robot	2.7	120
PUMA robot	3.1	40

Nippondenso is capable of producing a number of different automobile instruments on short notice with almost no inventory due to their highly flexible lines. Another Japanese manufacturer which has developed its own robots for production is Yamaha Motor Corporation [131]. They have developed the CAME (Computer-Aided Manufacturing Equipment for assembly operation) robot with the following specifications:

- Capable of handling material
- Capable of feeding parts
- Capable of assembling parts

- Has set up times less than 1 cycle long (20 - 60 sec)
- Operates at speeds as fast as humans
- Positional accuracy of < .1 mm
- Weighs about the same as a man
- Same cost as yearly salary of an operator

Yamaha has used these robots with multi-level vibratory bowl feeders predominantly for motorcycle engine assembly.

Ranky describes a project which is sponsored for the most part by the Hungarian government and is implemented at the Cspel Machine Tool company [158]. The system uses a direct access part handling robot versus an in line system. The robot can choose from a number of different process sequences and has direct access to any machining or inspection station. A fairly sophisticated flexible manufacturing system is currently being operated at the Fujitsu-Fanuc plant in Japan [131]. The plant produces robots and small machine tools using unmanned machining. Robots load the machine tools and machined parts and raw materials are carried from station to station by wire guided carts. Presently, all assembly at the plant is being done manually.

Sony has developed the FX-1 assembly system which assembles 50 percent of the parts in the drive mechanism for the Walkman II [131]. The system consists of X-Y tables which position pallets containing parts and assembly areas for four Walkman II assemblies. Unidirectional insertion is performed by single degree of freedom actuators which may be fitted with any number of end effectors. The pallets which are molded plastic trays are manually loaded before entering the system. Conveyers move pallets in and out of the assembly stations. The system can easily tolerate changes in both model design (by remolding the part trays) and production quantity. Other attributes of the system are 2 second cycle time, 0.015mm accuracy, an average reprogramming time of one minute, and a production rate of 500 sets an hour (48 pieces / set). The Daini Seikosha Co., Ltd. (manufacturer of Seiko watches) have been producing watches with almost no manual intervention for about 10 years [131]. They use rotary and in line machining and assembly centers for the production of mechanical and IC quartz watches. There are 6 lines utilized for the production of 10 models. 5 people fix jams during the second shift, which is the production shift, and 80 people maintain 300 machines during the first shift. There is no central monitoring or diagnostics and the machines are capable of positional accuracies smaller than .0004 inches (10 microns) and production quantities of 200,000 watch baseplates per month.

Automated Factories of the Future

Continuing investment in automated manufacturing equipment will result in continuing improvements in productivity. There will be a corresponding change in the nature of many existing jobs on the manufacturing floor. Possibly more significantly, as more tasks are automated, manufacturing plants will run in more structured ways. This will result in a reduction in the need for support personel. As a factory's performance more closely resembles performance of a computer model of the factory, more of the jobs involving flow of information will be performed by computer. The long term result will be a restructuring of the manufacturing environment.

2.2 Classification of Assembly Operations

Successful flexible assembly systems will come about through new developments in robots, versatile peripheral hardware, and more efficient cell programming techniques. In order to design useful systems, it is first instructive to study common mechanical assembly operations. A study at the Charles Stark Draper Laboratory [104,133] lists the twelve most common assembly operations in ten products. Researchers at the Draper Labs found that about 33 percent of the operations studied are peg-in-hole operations, 27 percent are screw insertions, and 12 percent are push and twist operations. It was also found that most of the operations occurred from the same direction of the assembly (e.g. 80 percent of all peg in hole operations were from the same direction).

When considering the set of capabilities which automatic assembly machines should exhibit, one needs to consider that the operations cited in the Draper study as well as the present study were from products which were designed for human assembly. Integration of the process and product design may produce products with a relatively small set of required assembly operations which are capable of being assembled with present state of the art machines. Extension of the capability of these machines will generate more permissible assembly operations for the product designer's consideration. Thus, the compatibility between product and flexible assembly system is dynamic and is extendable through an iterative design process between the two elements.

2.2.1 Assembly Task Analysis

The operations necessary to perform the assembly of 7 consumer and automotive products were studied. The investigation was carried out to

- investigate which mechanical assembly operations are prevalent in certain types of products,
- determine which operations can and cannot be accomplished by an unaided six degree-of-freedom manipulator,
- investigate the difficulty of the different operations.

None of the products were machine assembled, nor was their design optimized for ease of assembly (i.e. they were not designed for automated assembly [31,33]). Ten of the most prevalent mechanical assembly operations were identified:

1 - *Unstable Assembly*: Any operation where a part will not maintain its proper position under just the force of gravity. A plate without fasteners covering a long, thin compression spring is an example.

2 - *Required Orientation of Another Part Prior to Assembly*: Stabilizing (fixing the position) of an already assembled part prior to insertion of a new part.

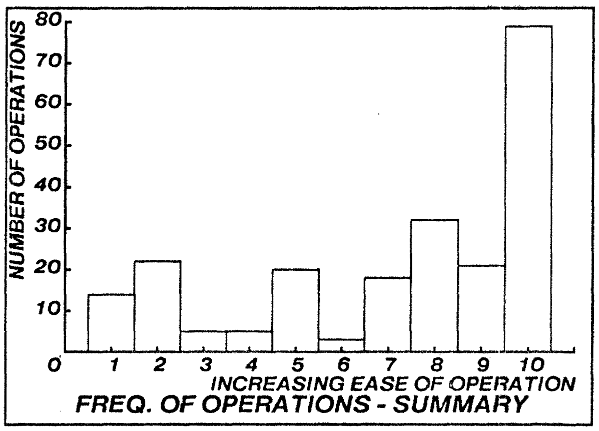
3 - *Retaining Clip Insertion*: includes assembly of internal and external snap rings and "E" clips.

4 - *Spring Insertion/Compression*: Operations which require insertion of parts which must be mechanically stressed prior to their installation.

5 - *Plastic Heading*: Heading of rivets and other fastening techniques requiring plastic deformation of material.

6 - *Unstable Inversion*: Requires that a part or assembly of parts be reoriented prior to assembly such that without constraint, they would become unstable and fall apart.

7 - *Non-screw Twisting*: Includes all helical insertions which are not performed with standard screws.



1) Unstable Assembly
2) Orient Another Part
3) Clip Insertion
4) Compress Spring
5) Plastically Deform
6) Unstable Inversion
7) Non-Standard Screwing
8) Press Fit
9) Standard Screwing
10) Unidirectional Insertion

Figure 2.1: Summary of frequency of assembly operations for products studied.

8 - Press Fit: Similar to *unidirectional insertions* except there is an interference fit rather than a clearance.

9 - Screw Insertion: Driving of standard shaped screws only. Specially designed parts which are screwed into an assembly are not included in this classification.

10 - Unidirectional Insertion: Any unidirectional insertion with a clearance fit. There is no restriction on the geometric form of the parts so long as the parts are rigid and the insertion direction is a straight line.

The frequency of occurrence of these operations for a number of automotive parts and consumer electromechanical products is summarized in Figure 2.1. No electrical component assembly operations (wiring, switch connection, printer circuit board component assembly, etc.) were included in this classification. The operations are listed along the abscissa in order of increasing ease of task completion for a single armed robot using only very simple tools. A subjective rating system was used for comparisons.

The simplest operation for a single manipulator is the *unidirectional insertion* operation where parts may be assembled in a straight line fashion and no sensing other than manipulator position feedback is required. This operation is the most prevalent comprising about 36 percent of all assembly operations in the assemblies studied. The *unidirectional insertion* operation is similar but slightly more comprehensive than the Draper peg-in-hole classification. The 36 percent frequency of occurrence seems to be in agreement with the 33 percent frequency reported for peg-in-hole operation in the Draper study. *Screwing* and *Non-standard Screwing* operations comprised about 18 percent of the operations studied. This is a bit less than the 27 percent reported by the Draper Labs. A significance test showed that not enough data was taken and too few assemblies were analyzed in either of the studies to allow meaningful comparisons to be made.

Operations 7 and 10 in Figure 2.1 may usually be accomplished with an unaided six degree of freedom manipulator. Fewer degrees of freedom may often be sufficient (e.g. a SCARA robot successfully performs many assembly operations with three or four degrees of freedom). A robot with a limited rotation wrist can usually perform screwing operations successfully, but laboriously [153]. Operations 3, 4, 5, 8, and 9 are best performed by a manipulator with the assistance of a special tool. Although operations 1, 2, and 6 may be accomplished with two or more manipulators, in many cases they may be performed with less complexity using a single manipulator and a relatively simple auxiliary device.

2.3 Peg-in-Hole Assembly Failure Modes

The most frequent assembly operation according to the Draper Lab study is the peg in hole insertion. Both two-dimensional [201] and three-dimensional [38,80] peg-in-hole tasks have been studied in detail. The results from some of the two dimensional studies are included here. The two-dimensional results approximate the results for the three-dimensional cylindrical peg-in-hole with small clearances. The two-dimensional analysis is also accurate for the rectangular peg-in-hole case when the rotation errors are about one of the bottom edges of the peg. More complicated interactions between rectangular peg and hole occur with arbitrary misalignments [38].

Three modes of failure for a peg-in-hole assembly are considered (see Figure 2.2).

Chamfer crossing failure: Initial translational alignment of parts is not within the range defined by the chamfers.

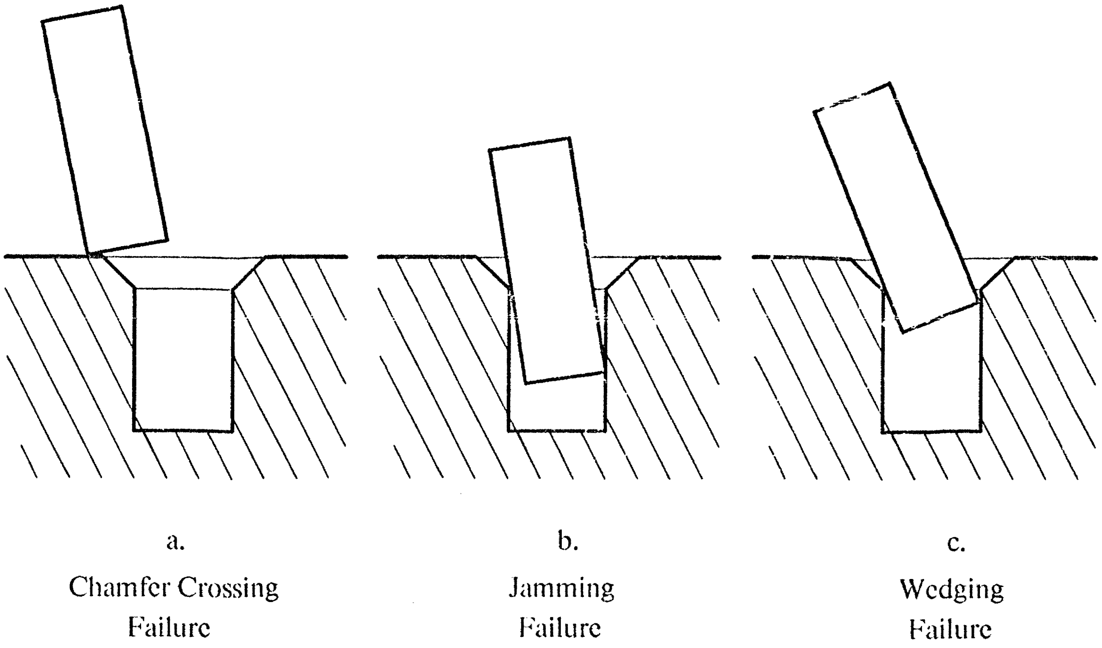


Figure 2.2: Failure modes for the peg in hole assembly. a. Initial translational alignment is not within chamfer limits. b. Jamming occurs. c. Wedging occurs.

Jamming failure: The insertion force is not in the proper direction to overcome the friction during two point contact.

Wedging failure: The initial misalignments and/or coefficient of friction are large enough such that the forces generated during two point contact will always equal the applied insertion force. When wedging occurs, often the only way to proceed with the assembly without damaging the parts is to reverse the direction of the insertion.

Failure modes involving jamming during one point contact and friction induced reaction forces during chamfer crossing are not considered here.

The chamfer crossing failure mode may be overcome by increasing the accuracy of the part alignment or by increasing the chamfer size of the parts, although such a part design change may not always be possible. The jamming failure mode may be overcome by applying a larger insertion force along the hole axis without proportionally increasing forces in the perpendicular directions. Many mechanical manipulators can apply forces which are much greater than those needed to

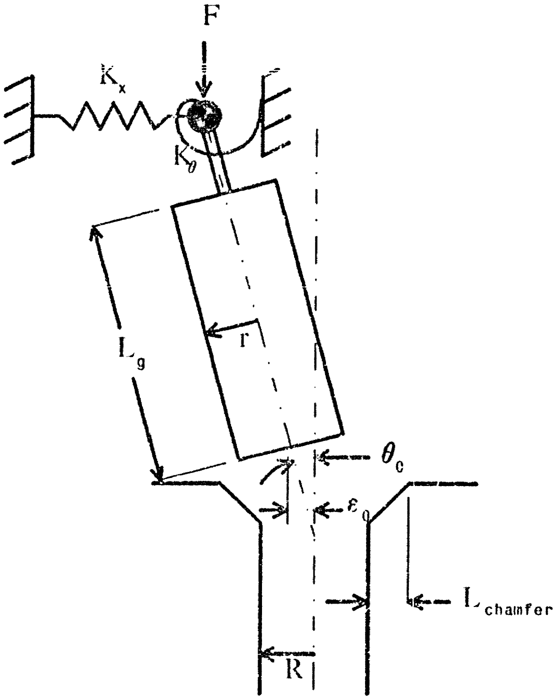


Figure 2.3: Parameters for the two-dimensional peg in hole assembly (from [201]).

overcome jamming. The chamfer crossing and wedging failure modes are often the most constraining and will be used to quantitatively evaluate the performance of assembly systems.

The criterion for avoiding the above failure modes are derived in [201]. The criterion for chamfer crossing and wedging are included here (refer to Figure 2.3).

- To cross the chamfer we need

$$|\epsilon_0| < L_{chamfer} \tag{2.1}$$

- To avoid wedging we must have

$$|\theta_0 + s\epsilon_0| < \frac{c}{\mu} \tag{2.2}$$

where

$$s = \frac{L_g}{L_g^2 + K_\theta / K_x},$$

μ is the coefficient of friction and c is the clearance ratio

$$c = \frac{R - r}{R}.$$

Equations (2.1) and (2.2) give an analytical technique for determining the limits of the assembly tasks which can be performed by an assembly system based upon the translational and orientation errors of the system.

2.4 Manipulator Repeatability, Accuracy and Local Accuracy

In this section, definitions of terms related to the accuracy of a manipulator are reviewed.

What Should be Measured?

Repeatability and accuracy are each often specified by a single number in the literature; presumably the maximum value. Since the location of the manipulator near some specified position can be described as a random vector (see Section 2.7.3), the specification for these errors should be given as moments (actually sample statistics) of the probability density of the components of the vector. Since it is often reasonable to assume that the distribution is approximately Gaussian shaped and that it is symmetric in all directions, it is sufficient to give just the second moment to describe the stochastic behavior about the mean. The sample standard deviation [53] is the metric used in this thesis.

Repeatability and accuracy should be specified for all degrees of freedom. Often robot specifications from the manufacturer only include translational repeatability.

Robot Repeatability

Robot repeatability is defined to be the capability of the robot to return to a previously visited location; that is, a particular location where the joint angles were recorded. The robot may return to this position using any path in the workspace. Some other definitions of repeatability assume that the manipulator always approaches from the same direction.

Repeatability is usually measured by making a number of readings the location of the manipulator after it has moved to a particular position. An accurate measuring device such as a theodolite [117,203], a set of dial gages [186] or a part position sensor such as the one described in Chapter 4 is used to take the measurement. In between measurements, the manipulator should move each of its joints through a significant fraction of their full range. During an assembly procedure, the robot might approach a previously specified position using a different path from that used during the teaching process. It might also execute a different path to contend with certain part misalignments. Thus, for an accurate repeatability measure, the test should entail approaches through different paths.

Robot Accuracy

Accuracy is the ability of a manipulator to move to a specified position in its world coordinate frame. This is a difficult quantity to measure because the actual location of the world coordinate frame with respect to observable robot frames is usually not known precisely. A good approximation to the accuracy measurement may be obtained using a sensing system which can measure the position of the end effector of the robot with respect to an arbitrary coordinate frame throughout the workspace. A "best fit" world frame may then be found from data taken over the entire workspace [182]. An absolute error may then be calculated from this approximate world frame.

Local Accuracy

Local accuracy is the accuracy of the manipulator within a limited volume with respect to an arbitrary base coordinate frame. This base frame is often the center of the specified volume of interest. This specification is more appropriate in a position-sensor-based assembly system since measurements and corrections are always made within a small volume about a nominal position. Relatively large inaccuracies which might occur near the bounds of the workspace do not affect a measurement of the local accuracy near the center of the workspace.

Step	Task
0.	**Teach** task, **calibrate** system.
1.	**Feed** and precisely orient part.
2.	**Acquire** part.
3.	**Move** to mating approach position.
4.	**Mate** part.

Table 2.1: Assembly procedure for the *a priori suppression* method of error elimination.

2.5 Assembly Procedure Using the *A Priori* Error Suppression Method

As an assembly system which relies solely on the precise location of the parts and the precise motion of the manipulator, the assembly procedure for the *a priori suppression* method is deterministic; that is, it is not altered by the state of the system. Since all of the part locations are precisely known, the job of the manipulator is to go to one of these positions, grasp the part, reposition it over the assembly and mate it. Table 2.1 gives the assembly procedure for the *a priori suppression* technique. The method used to teach the task to the manipulator is usually the "teach by showing" method. In this teaching method, the operator digitizes robot positions by positioning the manipulator with a teach box.

Alternatively, an offline programming technique may be used to teach the task to the robot (see References [29,114] for overviews). In this teaching method, a model of the workspace is stored in the computer. The assembly sequence is input by the user and the task is either automatically or manually generated. A manually generated task usually involves a user interacting with a computer aided design and graphics system, planning manipulator motions and checking for interference. Present offline programming systems do not model certain physical phenomena and do not offer the programmer much assistance with some of the more important issues in planning the assembly. Since forces between parts, friction, and dimensional tolerances are not usually modeled, assembly failure modes cannot be predicted.

Some experimental offline programming systems attempt to model some important physical interactions. Part clearances and tolerances are taken into account in [34,189] and planning of fine motion with friction is dealt with in [36,115]. Some authors have dealt with the grasp planning issue [142,164].

In order that the actual assembly environment conform closely to the model, the elements of the assembly system must be referenced to the world coordinate frame. This may either be done by accurately jigging the components or by having the manipulator calibrate their location.

2.6 Assembly Procedure Using the Measurement and Removal Method

A sensor-driven assembly system relies on a sensing device to determine the location of certain part features with a precision sufficient for required assembly tasks. A sketch of such a system is shown in Figure 2.4. The feature sensing occurs just prior to the part being mated to the assembly. In order to avoid specialized fixturing, reduce the amount of uncertainty in the position of the mating part, and reduce user introduced teaching errors, it may be desirable to equip a system with two part position sensors; one sensor to measure the location of the part in the assembly and one to measure to position of the part in the robot end effector (in some cases it may be possible to use a single sensor for both functions).

Manipulator mounted vision sensors have previously been used in an attempt to locate parts prior to grasping them [4,23]. Because significant part positioning errors may occur during part grasping, part sensing should occur after the part has been firmly grasped. A stationary sensor mounting was chosen over a manipulator mounting for the following reasons.

- The best features to sense are the mating features of parts. If the sensor were mounted on the manipulator, the mating feature may be difficult to sense since it will most likely face away from the upper part of the robot arm.

- With a manipulator mounted sensor there is limited flexibility in part orientation during sensing. Only the joints between the sensor mount and the end effector are available for reorientation prior to sensing. Additional degrees

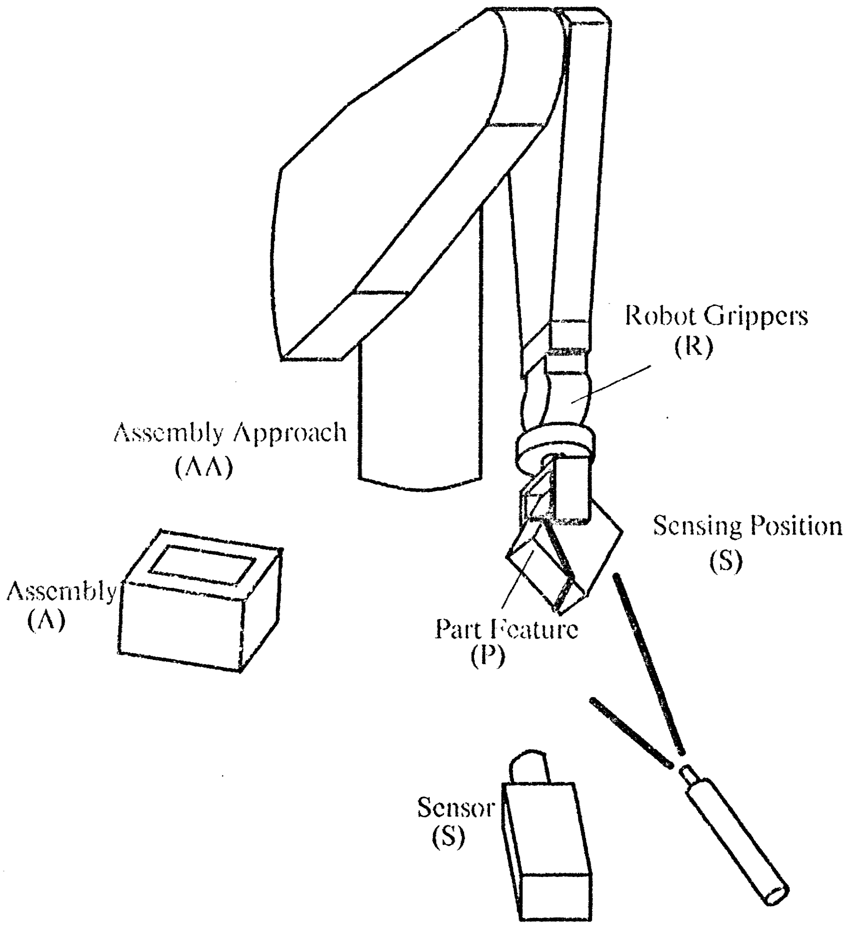


Figure 2.4: Sensor-driven assembly system.

of freedom would be required for arbitrary part positioning relative to the sensor.

- If the measurement is made relative to the world frame rather than to the robot frame, the positioning of the part in the assembly is less dependent on the calibration between the sensor and the manipulator. This means that the manipulator may be moved slightly or substituted with an entirely different manipulator without having to reteach the assembly task.

2.6.1 Assembly Procedures for Systems Using One and Two Sensors

The basic assembly system consists of one or two part position sensors, a six degree-of-freedom mechanical manipulator, an end effector which can firmly grasp all required parts, and a part orientation and delivery system. It is not required that the end effector fixture the position of a part, merely secure it so there is no relative motion between it and the last link of the manipulator. Only approximate part orientation is required at the feeder. The precision of the feeding-orientation device depends upon the size of the sensing volume of the part position sensor. If only a single part position sensor is used, an assembly jig which locates the base part of the device being assembled is also required.

Table 2.2a lists the steps involved in a typical sensor-driven assembly task with a single sensor and Table 2.2b describes the procedure for a system with two sensors. A calibration procedure must first be performed to find the transformation between each sensor coordinate system and the robot coordinate system.

For the single sensor system, the actual assembly procedure is also preceded by a teaching session where the user digitizes two nominal manipulator positions: a position which aligns the part with the assembly and a position in the vicinity of the active sensor volume. Alignment to the assembly is performed by either "eyeballing," using gaging instruments, or by guiding the manipulator through trial-and-error insertions. During teaching, the sensor system records a nominal feature location with the part positioned at a sensing location. This sensor reading is used as a baseline reading for subsequent measurements. The baseline reading is the "correct" sensor reading for the system to assemble the part using the nominal program learned during teaching. During an assembly task, commanded robot

Step	Task
0.	**Teach** task, **calibrate** system.
1.	**Feed** and grossly orient part.
2.	**Acquire** part.
3.	**Position** to nominal sensor position.
4.	**Sense** part feature.
5.	**Move** to corrected approach position.
6.	**Mate** part using corrected path.

a.

Step	Task
0.	**Calibrate** both sensors to robot.
1.	**Sense** part feature in assembly.
2.	**Feed** and grossly orient part.
3.	**Acquire** part.
4.	**Position** to nominal sensor position.
5.	**Sense** part feature in end effector.
6.	**Move** to corrected approach position.
7.	**Mate** part using corrected path.

b.

Table 2.2: Assembly procedures for a typical sensor-driven assembly tasks. a. Procedure for a single sensor system, b. Procedure for a dual sensor system,

positions are calculated based on differences between a current reading and the baseline reading. The calculated robot positions create small alterations to the nominal robot program which cause part misalignments to be nullified.

In the dual sensor system, the task need never be directly taught to the manipulator. When the mating features are sensed by the two sensors, the necessary part reorientation may be directly calculated from the sensor data and the sensor-robot calibration.

For a system with a single part position sensor, the assembly procedure starts with the robot acquiring the part and bringing it to the sensing position. The sensor takes a reading and while the computer is processing the image, the manipulator is free to transfer the part to the vicinity of the assembly. After processing the sensed information, the computer calculates the transformation from the sensed feature position to the previously recorded feature position. The robot is then instructed to execute the transformation which reorients the part for assembly.

In a system with two part position sensors, the part in the assembly is sensed in the first step. Depending upon the cycle time and the processing time, it may be possible to use a single set of image processing hardware for both images.

2.6.2 Coordinate Frame Definitions

The following are abbreviations used to specify the location of a particular coordinate frame of the assembly system (Figure 2.4 shows the approximate location of these frames). The convention for representing rigid transformations is given in Section 2.7.2.

W: The world frame. This is the world coordinate frame of the robot.

WA: This is the apparent world frame as defined by the robot's axes near the assembly position.

R: The robot coordinate frame. This is the frame associated with the robot grippers.

A: The assembly frame. This is the frame associated with the mating part in the assembly. It is assumed to be fixed with respect to the world frame.

AA: The assembly approach frame. This is the frame associated with the position just above the mating part. It is typically associated with the position of the robot when it is holding the part just over the assembly.

S: The sensor frame. This frame is associated with the image plane of the sensor camera. It is fixed with respect to the world frame. It also refers to the robot location in the vicinity of the active sensor volume.

P: The part frame. This is fixed to the part and is located at the feature to be sensed.

2.6.3 Correcting Sensed Misalignments

The system's knowledge of the required task is derived from the teaching phase. Deviations in part position from those defined during the teaching phase must be compensated by altering the robot's path. In this section, the transformation which aligns the part to the mating part is calculated.

The position of the sensed part feature with respect to the gripper frame is (see Figure 2.5)

$$ {}^{R}\mathbf{T}_{P} = \mathbf{T}_{R_S}^{-1}\ {}^{S}\mathbf{T}_{W}^{calib\ -1}\ {}^{S}\mathbf{T}_{P_S} \tag{2.3}$$

where ${}^{S}\mathbf{T}_{W}^{calib}$ is the calibrated transformation from the sensor frame to the world frame, ${}^{S}\mathbf{T}_{P_S}$ is the part feature position with respect to the sensor and $\mathbf{T}_{R_S}$ is the robot position while sensing is taking place. The robot position which will move the part back to the nominal sensing position, ${}^{S}\mathbf{T}_{P_S}^{teach}$ is

$$\begin{aligned} \mathbf{T}_{R_S,corr}^{-1} &= {}^{S}\mathbf{T}_{W}^{calib\ -1}\ {}^{S}\mathbf{T}_{P_S}^{teach}\ {}^{R}\mathbf{T}_{P}^{-1} \\ &= {}^{S}\mathbf{T}_{W}^{calib\ -1}\mathbf{T}_{P_{act,teach}}\ {}^{S}\mathbf{T}_{W}^{calib}\mathbf{T}_{R_S} \end{aligned} \tag{2.4}$$

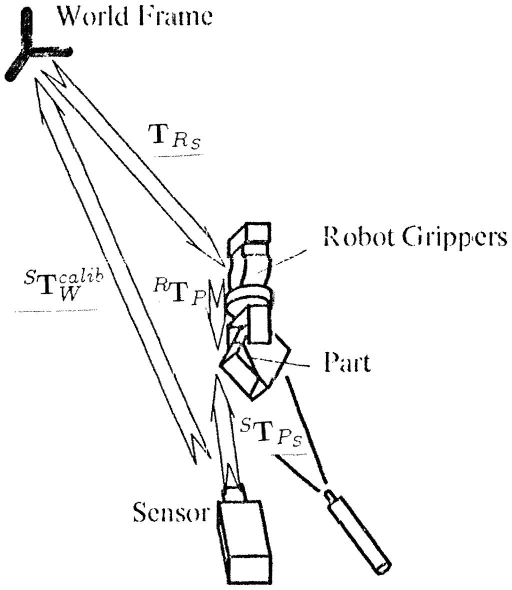


Figure 2.5: Calculation of the robot to part transformation.

where

$$\mathbf{T}_{P_{act,teach}} = {}^{S}\mathbf{T}^{teach}_{P_S}\, {}^{S}\mathbf{T}^{-1}_{P_S}$$

and ${}^{S}\mathbf{T}^{teach}_{P_S}$ is the nominal feature position with respect to the sensor during teaching.

When the robot "shows" the part to the sensor during the execution of a task, it moves to the same sensing position as the one during the teaching phase ($\mathbf{T}_{R_S} = \mathbf{T}^{teach}_{R_S}$). Thus, the corrected robot position near the assembly approach position is

$$\begin{aligned} \mathbf{T}_{R_{AA}} &= \mathbf{T}^{teach}_{R_{S,AA}}\, \mathbf{T}^{-1}_{R_{S,corr}} \\ &= \mathbf{T}^{teach}_{R_{S,AA}}\, {}^{S}\mathbf{T}^{calib\,-1}_{W}\, {}^{S}\mathbf{T}_{P_{art,teach}}\, {}^{S}\mathbf{T}^{calib}_{W}\, \mathbf{T}^{teach}_{R_S} \end{aligned} \qquad (2.5)$$

where $\mathbf{T}^{teach}_{R_{S,AA}}$ is the command transform from the sensing position to the assembly approach position during the teaching session. Equation (2.5) gives the corrected assembly approach position as a function of the deviation of the location of the part from the taught position.

2.7 Classification and Analysis of Errors

In this section methods for predicting the errors in an assembly system are presented in order to determine the specifications for a part position sensor. First a calculus for manipulating position errors is discussed, then errors from propagation of uncertainties in initial part positioning, robot motion commands, internal kinematic models, sensor readings, and sensor-frame to manipulator-frame calibration are explored.

2.7.1 Assumptions

In order to simplify the analysis we make the following assumptions.

- Robot position is specified to (not read from) joint encoders whenever possible. This avoids doubling the robot repeatability error.

- All errors are small and Gaussian (where applicable).

- Inaccuracies in the robot's internal kinematic model (differences between commanded and actual motion) may be accurately modeled by a transformation error (small rigid rotation and displacement) in its world coordinate frame and a finite robot repeatability.

A robot independent representation is used and exact kinematics are not modeled; thus, only approximate dependencies may be examined with this error model. The magnitudes of position errors as a function of the position of the three joints of a spherical wrist are analyzed in Appendix D.

2.7.2 Representation of Rigid Transformations

Homogeneous transformation matrices (4×4 matrices – see [147]) are used to represent rigid transformations and are denoted by the boldface letter **T**. The subscript denotes the object that the transform refers to. The second level of subscript signifies the start and end region specified by the transform. The optional left superscript denotes the reference frame in which the transform is defined. This superscript defaults to the world frame (W). For example, the transformation of a

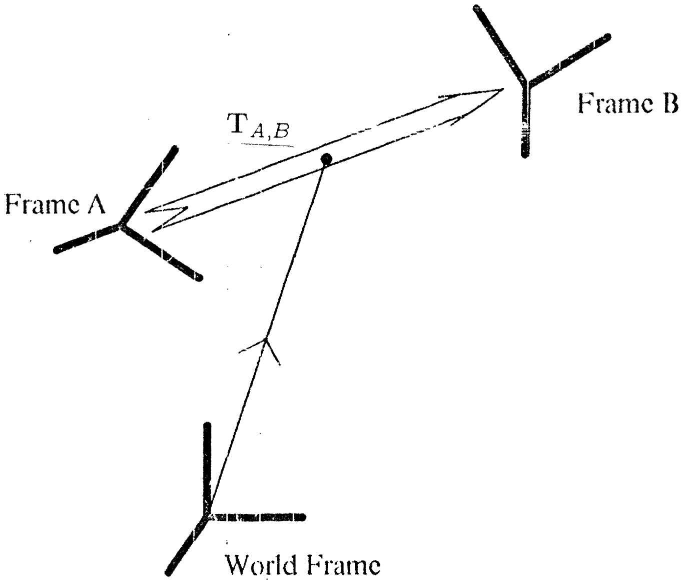


Figure 2.6: Graphical representation of a rigid transformation. The transformation shown, $\mathbf{T}_{A,B}$, is from frame A (six coordinates) to frame B. The extra line from the world frame to the transformation indicates that the transformation is specified with respect to this frame.

part from the sensing position to the assembly approach position with respect to the sensor frame is ${}^{S}\mathbf{T}_{P_{S,AA}}$.

A rigid transformation may also be thought of as a vector in six space starting at a set of initial coordinates (3 translations and 3 rotations) and ending at a set of final coordinates. We may graphically represent a rigid transformation in a three-dimensional subspace (say the space defined by the translation coordinates x, y, and z) projected into two-dimensions. This is a transformation graph. In order to differentiate between a transformation and a 3-vector, the transformation is drawn as a double arrow (see Figure 2.6). An extra line extends from some point in the subspace to the transformation which indicates the frame from which the transformation is specified. If the transformation is defined in the frame associated with the starting coordinates, the extra line need not be drawn. A certain transformation may be found from a transformation graph by tracing through the graph. Tracing backwards over a transformation means that the inverse of the transformation should be used; thus, the transformation from B to A in the graph in Figure 2.6 is $\mathbf{T}_{A,B}^{-1}$.

2.7.3 Representation of Errors

Errors in a transformation are composed of a displacement error and a rotation error, each having a magnitude and a direction; thus, we can represent each of them them as a random vector[1]

$$\tilde{\epsilon} = \left\{ \begin{array}{c} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{array} \right\}, \quad \tilde{\alpha} = \left\{ \begin{array}{c} \alpha_x \\ \alpha_y \\ \alpha_z \end{array} \right\}.$$

An element of the displacement error vector, $\tilde{\epsilon}$, is just the cartesian error (distance from the origin) times the appropriate direction cosine of the unit vector pointing in the direction of the error. An element of the rotation error vector, $\tilde{\alpha}$, is the total rotation error angle times the appropriate direction cosine of the unit vector pointing in the direction of the axis of rotation.

We now define a random transformation matrix, $\Delta\mathbf{T}$, as the homogeneous transformation matrix which is associated with a random displacement error, $\tilde{\epsilon}$, and a random rotation error, $\tilde{\alpha}$. This transformation is a function of a vector of six variables

$$\tilde{\delta T} = \left\{ \begin{array}{c} \tilde{\epsilon} \\ \cdots \\ \tilde{\alpha} \end{array} \right\}.$$

The statistics of the random transformation matrix are governed by the six dimensional joint probability density function (PDF) $p_{\tilde{\delta T}}(\tilde{\delta T}_0)$. This density function gives the distribution of probability for the components of the rotation and displacement error vectors. A sample joint three-dimensional PDF is sketched in Figure 2.7. The density of the "cloud" represents the probability density. In general the joint PDF's for transformation errors have a zero mean (the expected value for the error vector is $E[\tilde{\delta T} = [0\,0\,0\,0\,0\,0]^T)$.

2.7.4 Combining Errors From Independent Sources

The PDF of a random transformation error which is the sum of N independent random transformation errors

$$\tilde{\delta T} = \tilde{\delta T}_1 + \tilde{\delta T}_2 + \cdots + \tilde{\delta T}_N$$

[1] A random vector is a one dimensional matrix of random variables. It is denoted by a ˜ over a symbol.

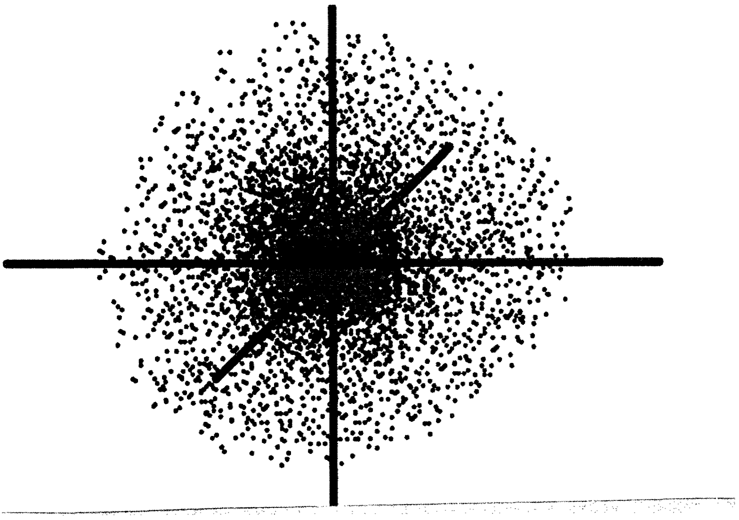


Figure 2.7: Three dimensional probability density function.

may be found by convolving the component PDF's [53]

$$p_{\delta\tilde{T}}(\delta\tilde{T}_0) = p_{\delta\tilde{T}_1}(\delta\tilde{T}_{1,0}) \otimes p_{\delta\tilde{T}_2}(\delta\tilde{T}_{2,0}) \otimes \cdots \otimes p_{\delta\tilde{T}_N}(\delta\tilde{T}_{N,0}) \tag{2.6}$$

where $\otimes$ is the convolution operator. Figure 2.8 shows a sketch of the result of convolving two independent planar displacement error PDF's.

Simplifications for Axisymetric Probability Density Functions

If the component PDF's are symmetric in ϵ_x, ϵ_y, and ϵ_z and also symmetric in α_x, α_y and α_z, then rigidly transformed PDF's are identical to the untransformed PDF's (see Section 2.7.7).

Transformation of Errors in Different Reference Frames

In order to combine their PDF's, all transformation errors must be specified in the same coordinate system. If the errors are specified in different coordinate systems, the PDF's must undergo a coordinate transformation to bring them to a base coordinate system. For the case of N error transformations each specified in a different frame, $\mathcal{F}_1, \ldots, \mathcal{F}_N$

$$^{\mathcal{F}_1}\delta\tilde{T}_1,\ ^{\mathcal{F}_2}\delta\tilde{T}_2,\ \ldots,\ ^{\mathcal{F}_N}\delta\tilde{T}_N$$

we must first transform them to the same frame to obtain

$$\delta\tilde{T} = {}^{\mathcal{F}_1}\delta\tilde{T}_1 + {}^{\mathcal{F}_1}\delta\tilde{T}_2 + \cdots + {}^{\mathcal{F}_1}\delta\tilde{T}_N \tag{2.7}$$

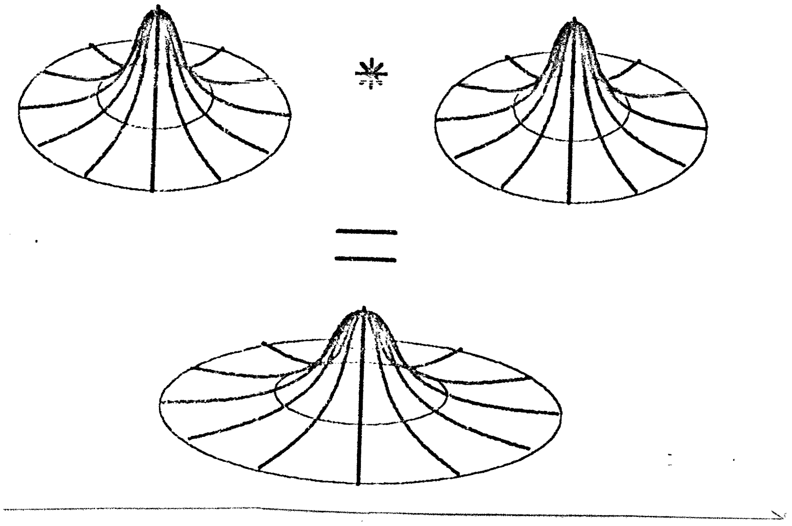


Figure 2.8: Result of convolving two two-dimensional probability density functions.

before convolving their PDF's. If, however, the PDF's are symmetric, the errors may be combined without transforming to the same coordinate system

$$\tilde{\delta T} = {}^{\mathcal{F}_1}\tilde{\delta T}_1 + {}^{\mathcal{F}_2}\tilde{\delta T}_2 + \cdots + {}^{\mathcal{F}_N}\tilde{\delta T}_N. \tag{2.8}$$

Gaussian Distributed Errors

If the PDF's of components of a sum of transformation error matrices are Gaussian; that is, of the form

$$p_{\tilde{\delta T}}(\tilde{\delta T}_0) = \frac{1}{(2\pi)^3 |\mathbf{P}|^{\frac{1}{2}}} \exp[-\frac{1}{2}\, \tilde{\delta T}_0^T \mathbf{P}^{-1} \tilde{\delta T}_0],$$

where **P** is the covariance matrix (diagonal for symmetric PDF's) defined by

$$E[\tilde{\delta T}_0 \tilde{\delta T}_0^T],$$

where E is the expectation operator, then the result of the convolution of N PDF's is a PDF which is also Gaussian. The width (standard deviation) of each dimension of the resulting Gaussian is the square root of the corresponding diagonal component of **P** or

$$\sigma_i = \sqrt{\sum_{j=1}^{N} \sigma_{i,j}^2},$$

where the index i is taken over the six dimensions of $\tilde{\delta T}$ and $\sigma_{i,j}$ is the width in the i^{th} dimension of the PDF of the j^{th} transform error.

2.7.5 Random Transformations

A random transformation, $\Delta\mathbf{T}$, is a homogeneous 4×4 matrix whose rotation and translation components are random variables (Section 4.1). It describes the difference between a coordinate frame with uncertainty, $\mathbf{T}_A$, and its mean, $\bar{\mathbf{T}}_A$,

$$\mathbf{T}_A = \Delta\mathbf{T}\bar{\mathbf{T}}_A \tag{2.9}$$

Aside from having random variable components, this description is slightly different from the differential transformation, $\boldsymbol{\Delta}$, described in Paul [147]. The relationship between a random transformation, $\Delta\mathbf{T}$, and Paul's differential transformation, $\boldsymbol{\Delta}$, is

$$\boldsymbol{\Delta} = \Delta\mathbf{T} - \mathbf{I}$$

where $\mathbf{I}$ is the identity transform. The two representations have similar properties and many of the relationships derived in [147] are used here. The random transformation representation was chosen over the differential transformation representation because random transformations are homogeneous matrices and may be manipulated in the same manner as deterministic transformations.

2.7.6 Relative Random Transformations

A random transformation may describe the difference between a transformation and its mean in the global coordinate system as well as in a local coordinate system

$$\mathbf{T}_A = \bar{\mathbf{T}}_A\,{}^{A}\Delta\mathbf{T} \tag{2.10}$$

where ${}^{A}\Delta\mathbf{T}$ is relative to frame $\bar{\mathbf{T}}_A$.

The relationship between the relative random transformation and the global random transformation may be found by equating Equations (2.9) and (2.10)

$${}^{A}\Delta\mathbf{T} = \bar{\mathbf{T}}_A^{-1}(\Delta\mathbf{T})\bar{\mathbf{T}}_A. \tag{2.11}$$

This is a coordinate transformation of the random transformation from the world frame to the frame A. If we assume the mean transformation is of the form

$$\bar{\mathbf{T}}_A = \begin{bmatrix} \hat{n} & \hat{o} & \hat{a} & \vec{d} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where $\hat{n}$, $\hat{o}$, and $\hat{a}$ are unit vectors in the respective local x, y, and z directions and $\vec{d}$ is the vector from the origin of the world frame to the origin of the frame $\bar{\mathbf{T}}_A$, then the vector components of the relative random transformation with respect to the global components are (follows from Paul's derivation [147])

$$^{A}\tilde{\alpha} = \begin{Bmatrix} \hat{n} \cdot \tilde{\alpha} \\ \hat{o} \cdot \tilde{\alpha} \\ \hat{a} \cdot \tilde{\alpha} \end{Bmatrix} \tag{2.12}$$

$$^{A}\tilde{\epsilon} = \begin{Bmatrix} \hat{n} \cdot ((\tilde{\alpha} \times \vec{d}) + \tilde{\epsilon}) \\ \hat{o} \cdot ((\tilde{\alpha} \times \vec{d}) + \tilde{\epsilon}) \\ \hat{a} \cdot ((\tilde{\alpha} \times \vec{d}) + \tilde{\epsilon}) \end{Bmatrix}. \tag{2.13}$$

Alternatively, if we describe the transformation, $\bar{\mathbf{T}}_A$, as a rotation by an angle, θ, about an axis, $\hat{k}$, then a translation by a vector $\vec{d}$ then the rotation vector corresponding to the relative random matrix, $^{A}\Delta\mathbf{T}$ is

$$^{A}\tilde{\alpha} = \hat{k}(\hat{k} \cdot \tilde{\alpha}) + \sin(\theta)(\hat{k} \times \tilde{\alpha}) + \cos(\theta)[\tilde{\alpha} - \hat{k}(\hat{k} \cdot \tilde{\alpha})] \tag{2.14}$$

Thus the magnitude of the rotation error is the same but the direction of the original rotation error axis, $\frac{\tilde{\alpha}}{|\tilde{\alpha}|}$, has been rotated about the $\hat{k}$ axis by an angle θ.

The displacement component of $\Delta\mathbf{T}$ in vector form is

$$\begin{aligned} ^{A}\tilde{\epsilon} \;=\; & \hat{k}(\hat{k} \cdot \tilde{\epsilon}) + \sin(\theta)[\hat{k} \times \tilde{\epsilon}] + \cos(\theta)[\tilde{\epsilon} - \hat{k}(\hat{k} \cdot \tilde{\epsilon})] + \\ & (\tilde{\alpha} \cdot \hat{k})(\vec{d} \times \hat{k}) + \sin(\theta)[\hat{k}(\vec{d} \cdot \tilde{\alpha}) - \tilde{\alpha}(\vec{d} \cdot \hat{k})] + \cos(\theta)[\vec{d} \times (\tilde{\alpha} - (\tilde{\alpha} \cdot \hat{k})\hat{k})]. \end{aligned} \tag{2.15}$$

The first line of the above expression is the contribution of displacement error vector, $\tilde{\epsilon}$, after being rotated. The second line is the displacement due to the rotational uncertainty, $\tilde{\alpha}$, of the commanded coordinate frame.

Unlike the rotation error, the magnitude of the displacement error is a function of the direction of the frame $\bar{\mathbf{T}}_A$. The contribution from $^{A}\tilde{\epsilon}$ is a displacement of the same magnitude as $\tilde{\epsilon}$ but rotated about axis $\hat{k}$ by the angle $-\theta$. The contribution from $\tilde{\alpha}$ depends upon the relative directions between $\tilde{\alpha}$, $\hat{k}$, and $\vec{d}$. Thus, the magnitude of the displacement error of the relative transformation is function of the location of the original frame, $\bar{\mathbf{T}}_A$.

2.7.7 Identification of Sources of Error

Position errors in assembly systems are usually generated from sources in different locations in the workspace. The following sections analyze some manipulator dependent errors.

Errors due to Transformations of Frames with Uncertainty

Assume that a coordinate frame, $\mathbf{T}_A$ has an error associated with it

$$\mathbf{T}_A = \bar{\mathbf{T}}_A {}^A\Delta\mathbf{T}_A, \tag{2.16}$$

where $\bar{\mathbf{T}}_A$ is the mean (deterministic) frame and ${}^A\Delta\mathbf{T}_A$ is the homogeneous transformation matrix (probabilistic) describing the transformation of the actual with respect to the mean frame. ${}^A\Delta\mathbf{T}_A$ has a zero rotation and displacement mean (the expected value of the associated vector is $[0\ 0\ 0\ 0\ 0\ 0]^T$). Post multiplication implies that the transformation is carried out with respect to the mean frame, $\bar{\mathbf{T}}_A$.

We are interested in investigating the positional error after the frame $\mathbf{T}_A$ has undergone the commanded transformation $\bar{\mathbf{T}}_C$ (deterministic). The final position is

$$\begin{aligned}\mathbf{T}_F &= \bar{\mathbf{T}}_C\mathbf{T}_A \\ &= \bar{\mathbf{T}}_C\bar{\mathbf{T}}_A {}^A\Delta\mathbf{T}_A.\end{aligned} \tag{2.17}$$

The final mean position is

$$\bar{\mathbf{T}}_F = \bar{\mathbf{T}}_C\bar{\mathbf{T}}_A. \tag{2.18}$$

The error in the final position with respect to the mean final position is [from Equations (2.17) and (2.18)]

$$\begin{aligned}{}^F\Delta\mathbf{T}_F &= \bar{\mathbf{T}}_F^{-1}\mathbf{T}_F \\ &= {}^A\Delta\mathbf{T}_A.\end{aligned} \tag{2.19}$$

Thus, the magnitude of the relative error is insensitive to rigid transformations; however, the direction of the error vectors change with respect to the world frame (since each of the probabilistic error matrices are relative to the nominal transformation). Symmetric PDF's remain invariant through rigid transformations.

Errors Due to Moving Through Transformations with Uncertainty

Here we assume that the commanded transformation, $\mathbf{T}_C$, is in error

$$\mathbf{T}_C = \bar{\mathbf{T}}_C {}^C\Delta\mathbf{T}_C, \tag{2.20}$$

where $\bar{\mathbf{T}}_C$ is the mean (deterministic) transformation and ${}^C\Delta\mathbf{T}_C$ is the relative homogeneous transformation matrix (probabilistic) describing the transformation of the actual with respect to the mean commanded transformation. The final position of frame $\bar{\mathbf{T}}_A$ after undergoing transformation $\mathbf{T}_C$ is

$$\begin{aligned}\mathbf{T}_F &= \mathbf{T}_C\bar{\mathbf{T}}_A \\ &= \bar{\mathbf{T}}_C {}^C\Delta\mathbf{T}_C\bar{\mathbf{T}}_A.\end{aligned} \tag{2.21}$$

The error in the final position with respect to the mean final position is

$$\begin{aligned}{}^F\Delta\mathbf{T}_F &= \bar{\mathbf{T}}_F^{-1}\mathbf{T}_F \\ &= \bar{\mathbf{T}}_A^{-1} {}^C\Delta\mathbf{T}_C\bar{\mathbf{T}}_A.\end{aligned} \tag{2.22}$$

This is just a coordinate transformation through the transformation $\bar{\mathbf{T}}_A$. This makes sense since Equation (2.20) may be thought of as an error defined in the world frame, $\Delta\mathbf{T}_A = {}^C\Delta\mathbf{T}_C$, which is associated with the initial frame $\bar{\mathbf{T}}_A$.

Errors Due to Moving in a World Frame with Uncertainty

In this case we assume the world coordinate frame is in error. Errors in robot motion due to the robot's internal kinematic model may be modeled by errors in the world frame (see Figure 2.9).

$$\mathbf{T}_{W'} = \bar{\mathbf{T}}_W {}^W\Delta\mathbf{T}_W,$$

or

$${}^W\Delta\mathbf{T}_W = \bar{\mathbf{T}}_W^{-1}\mathbf{T}_{W'} \tag{2.23}$$

where $\bar{\mathbf{T}}_W$ is base world frame and $\mathbf{T}_{W'}$ is frame about which commands are executed and ${}^W\Delta\mathbf{T}_W$ is the relative transformation between the two. We have assumed that errors in the robot's internal kinematic model may be modeled as errors in the world frame. The desired final position, $\bar{\mathbf{T}}_F$, of a frame, $\bar{\mathbf{T}}_A$, after undergoing transformation $\bar{\mathbf{T}}_C$ is

$$\bar{\mathbf{T}}_F = \bar{\mathbf{T}}_C\bar{\mathbf{T}}_A \tag{2.24}$$

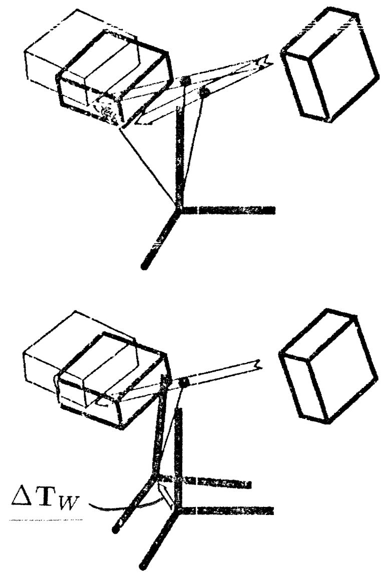


Figure 2.9: Errors in part positions due to manipulator errors may be equivalently represented by errors in the location of the world frame.

We are interested in errors due to the commanded transformation, $\bar{\mathbf{T}}_C$, being executed in frame $\mathbf{T}_{W'}$ rather than in frame $\bar{\mathbf{T}}_W$. First we find the location of frame $\bar{\mathbf{T}}_A$ with respect to the W' coordinate frame as

$$ {}^{W'}\mathbf{T}_A = {}^{W}\Delta\mathbf{T}_W^{-1}\bar{\mathbf{T}}_A \tag{2.25}$$

The final position after undergoing transformation $\bar{\mathbf{T}}_C$ in the W' coordinate frame is

$$\begin{aligned} {}^{W'}\mathbf{T}_F &= \bar{\mathbf{T}}_C{}^{W'}\mathbf{T}_A \\ &= \bar{\mathbf{T}}_C{}^{W}\Delta\mathbf{T}_W\bar{\mathbf{T}}_A \end{aligned} \tag{2.26}$$

The final position in the W coordinate frame is

$$\mathbf{T}_F = {}^{W}\Delta\mathbf{T}_W\bar{\mathbf{T}}_C\Delta\mathbf{T}_W^{-1}\bar{\mathbf{T}}_A. \tag{2.27}$$

The error in the final position with respect to the W frame is

$$\Delta\mathbf{T}_F = \bar{\mathbf{T}}_F^{-1}\mathbf{T}_F$$

$$= \bar{\mathbf{T}}_A^{-1}[\bar{\mathbf{T}}_C^{-1}{}^W\Delta\mathbf{T}_W\bar{\mathbf{T}}_C]^W\Delta\mathbf{T}_W^{-1}\bar{\mathbf{T}}_A. \tag{2.28}$$

This equation is essentially a nested form of Equation (2.22). The expression in the square brackets is of the same form of Equation (2.22) and generates a relatively small[2] transformation matrix when ${}^W\Delta\mathbf{T}_W$ is small. Postmultiplying this matrix by ${}^W\Delta\mathbf{T}_W^{-1}$ gives another small transformation matrix

$${}^W\Delta\mathbf{T}^*{}_W = [\bar{\mathbf{T}}_C^{-1}{}^W\Delta\mathbf{T}_W\bar{\mathbf{T}}_C]^W\Delta\bar{\mathbf{T}}_W^{-1}.$$

The equation then takes the form $\bar{\mathbf{T}}_A^{-1}{}^W\Delta\mathbf{T}^*{}_W\bar{\mathbf{T}}_A$ which is of the same form as Equation (2.22).

We now investigate errors which are deviations of a final position from a taught position rather than from an absolute position. In a position-sensor-driven assembly system, $\bar{\mathbf{T}}_A$ might be the position of the manipulator during the sensing phase and $\bar{\mathbf{T}}_C$ might be the commanded transformation which brings the parts into alignment. Thus, $\bar{\mathbf{T}}_C$ would vary depending on the part position in the grippers.

Still assuming a world coordinate frame with uncertainty, the final position during the teaching phase is [from Equation (2.27)]

$$\mathbf{T}_F^{teach} = {}^W\Delta\mathbf{T}_W\mathbf{T}_C^{teach}\Delta\mathbf{T}_W^{-1}\bar{\mathbf{T}}_A. \tag{2.29}$$

The final position during an assembly operation after sensing and calculating the corrected command transform is identically Equation (2.27) where $\bar{\mathbf{T}}_C$ is now the commanded transformation based on the sensed data. The difference between the taught final position and the actual final position is

$$\begin{aligned}\Delta\mathbf{T}_F &= \Delta\mathbf{T}_F^{teach\ -1}\mathbf{T}_F\\ &= \bar{\mathbf{T}}_A^{-1}{}^W\Delta\mathbf{T}_W\mathbf{T}_{C_{diff}}{}^W\Delta\mathbf{T}_W^{-1}\bar{\mathbf{T}}_A,\end{aligned} \tag{2.30}$$

where

$$\mathbf{T}_{C_{diff}} = \mathbf{T}_C^{teach\ -1}\bar{\mathbf{T}}_C.$$

The deviations from the taught positions are a function of errors in the world frame only if the commanded transformation, $\bar{\mathbf{T}}_C$, is different from the commanded

[2] A small transformation matrix is the identity transformation rotated by a small angle and shifted by a small displacement.

transformation during teaching, $\mathbf{T}_C^{teach}$. The effect of Equation (2.30) is to rotate the errors in ${}^W\Delta\mathbf{T}_W$ by the correction angle contained in $\mathbf{T}_{C_{diff}}$, and depending upon the magnitude of the correction, the resulting error may be quite small. In other words, if the part misalignments between the teaching phase and the measurement phase is relatively small, the accuracy in part positioning is insensitive to errors in the robot kinematic model.

If we assume that both the world frame uncertainty, ${}^W\Delta\mathbf{T}_W$, and the difference on commands, $\mathbf{T}_{C_{diff}}$, are random transformation matrices, we may obtain a PDF for the vector associated with the resulting final position error, $\Delta\mathbf{T}_F$.

The difference in the commanded transformation, $\mathbf{T}_{C_{diff}}$, in an assembly system (such as the one described in Section 6.6) is due to the part being in a different position in the gripper than the original position during teaching. We assume that this change in position is small (small angle approximations are valid) and the rotation associated with $\mathbf{T}_{C_{diff}}$ is described by a vector $\tilde{\theta}$; that is, the angle of rotation is $|\tilde{\theta}|$ and the axis of rotation is $\frac{\tilde{\theta}}{|\tilde{\theta}|}$. For $|\tilde{\theta}|$ small, the difference between a vector $\vec{x}$ and the rotation of $\vec{x}$ by $\tilde{\theta}$ is approximately $\tilde{\theta} \times \vec{x}$. We assume the displacement errors in ${}^W\Delta\mathbf{T}_W$ are described by a random vector, $\tilde{\epsilon}$, the rotation by a random vector, $\tilde{\alpha}$, and the displacement component of $\bar{\mathbf{T}}_A$ is $\vec{d}$ (deterministic). We also assume that the PDF's for $\tilde{\theta}$, $\tilde{\alpha}$, and $\tilde{\epsilon}$ are symmetric Gaussians with characteristic widths σ_θ, σ_α, and σ_ϵ respectively. It is helpful to note that the one dimensional marginal PDF for any component of a multi-variate Gaussian is Gaussian as well.

We are presently interested only in the errors from the uncertainty in the world frame, ${}^W\Delta\mathbf{T}_W$. In Equation (2.30), the start frame, $\bar{\mathbf{T}}_A$, is rotated by the negative (the inverse rotation) of the rotation error vector, $-\tilde{\alpha}$, then by the difference in part orientation, $\tilde{\theta}$, then by the rotation error, $\tilde{\alpha}$. When all rotations are small, the resultant rotation error is

$$\tilde{\alpha}_F = \tilde{\theta} \times \tilde{\alpha} \tag{2.31}$$

or

$$\tilde{\alpha}_F = \left\{ \begin{array}{c} \theta_2\alpha_3 - \theta_3\alpha_2 \\ \theta_3\alpha_1 - \theta_1\alpha_3 \\ \theta_1\alpha_2 - \theta_2\alpha_1 \end{array} \right\} \tag{2.32}$$

where θ_i, ϵ_i and α_i correspond to the i^{th} component of $\tilde{\theta}$, $\tilde{\epsilon}$ and $\tilde{\alpha}$ respectively. The total displacement from the last two matrices in Equation (2.30) is $\tilde{\epsilon} + \vec{d} + (\tilde{\alpha} \times \vec{d})$.

The final displacement error in Equation (2.30) due to the contributions of ${}^{W}\Delta\mathbf{T}_W$ (terms with $\tilde{\alpha}$ and $\tilde{\epsilon}$ only) is

$$\tilde{\epsilon}_F = \tilde{\theta} \times [\tilde{\epsilon} + (\tilde{\alpha} \times \vec{d})]. \tag{2.33}$$

or

$$\tilde{\epsilon}_F = \left\{ \begin{array}{l} \theta_2(\epsilon_3 + \alpha_1 d_2 - \alpha_2 d_1) - \theta_3(\epsilon_2 + \alpha_3 d_1 - \alpha_1 d_3) \\ \theta_3(\epsilon_1 + \alpha_2 d_3 - \alpha_3 d_2) - \theta_1(\epsilon_3 + \alpha_1 d_2 - \alpha_2 d_1) \\ \theta_1(\epsilon_2 + \alpha_3 d_1 - \alpha_1 d_3) - \theta_2(\epsilon_1 + \alpha_2 d_3 - \alpha_3 d_2) \end{array} \right\} \tag{2.34}$$

The PDF's of the components of $\tilde{\theta}$, $\tilde{\alpha}$ and $\tilde{\epsilon}$ are the marginal PDF's of uncoupled multivariate Gaussians which are independent Gaussian distributions. A number of terms in Equations (2.32) and (2.34) are products of two random variables. The PDF of the product of two Gaussian distributed independent random variables is a modified Bessel function of the second kind, order 0 (see Appendix E). Convolving a number of these Bessel functions gives a distribution which may be approximated by a Gaussian (central limit theorem [53]). We make the assumption that the resulting distributions of Equations (2.32) and (2.34) are Gaussians with covariance matrices $E(\tilde{\alpha}_F \tilde{\alpha}_F^T)$ and $E(\tilde{\epsilon}_F \tilde{\epsilon}_F^T)$ respectively. Since we have assumed all of the original distributions are symmetric and independent, the vectors of diagonal elements of the covariance matrices are

$$\vec{\sigma}^2_{\tilde{\alpha}_F} = \left\{ \begin{array}{c} 2\sigma_\theta^2\sigma_\alpha^2 \\ 2\sigma_\theta^2\sigma_\alpha^2 \\ 2\sigma_\theta^2\sigma_\alpha^2 \end{array} \right\} \tag{2.35}$$

and

$$\vec{\sigma}^2_{\tilde{\epsilon}_F} = \left\{ \begin{array}{c} \sigma_\theta^2[2\sigma_\epsilon^2 + (2d_1^2 + d_2^2 + d_3^2)\sigma_\alpha^2] \\ \sigma_\theta^2[2\sigma_\epsilon^2 + (d_1^2 + 2d_2^2 + d_3^2)\sigma_\alpha^2] \\ \sigma_\theta^2[2\sigma_\epsilon^2 + (d_1^2 + d_2^2 + 2d_3^2)\sigma_\alpha^2] \end{array} \right\}. \tag{2.36}$$

2.8 Errors in the *A Priori* Error Suppression Method

The sources of error which contribute to the misalignment of the parts include

- Positioning accuracy of the assembly jig.
- Inaccuracies in gripper constraint. These may be due to clearances between the gripper interface and the part, slop in the gripper mechanism, etc.

- Tolerance and clearance buildup in the parts already assembled.

- Robot repeatability.

- Teaching errors.

In an assembly system based on the *a priori* suppression method of error removal, parts are initially constrained in jigs. The locations of the jigs, and therefore the parts themselves, are referenced to some global frame in the assembly system. The positional uncertainty for parts constrained in jigs depends on the geometry of the parts and the method of constraint used by the jigs. To properly constrain parts with varying dimensions, the clearance between the parts and the jigs must be greater than the tolerances between the interfacing surfaces of the part. The clearances must also be large enough so that the parts do not jam during removal. Whether or not a part jams depends on the manner in which it is gripped as well as the jig clearances.

The function of the manipulator is to reposition the parts to an assembled position. In so doing it must retain the accurate position information provided by the part constraint system while altering the part positions. Part grasping is a critical phase where relatively large uncertainties in part location may be introduced. Uncertainties may be minimized by either fixturing parts in specially formed grippers or by maintaining the positional accuracy provided by the part jigs.

Experiments with the system described in Chapter 5 showed that it is difficult to maintain the accurate location of certain parts without using grippers which constrain the part's position. The act of grasping a part tends to displace the part slightly and cause it to apply a force to the constraining jig. Due to finite system compliance, once the part is free from the jig, this force may sometimes displace the part significantly. The displacement was found to be above the acceptable bounds for some of the assembly operations. A gripper which conforms to the shape of parts (self-conforming gripper) would maintain the position of the part without imparting unnecessary forces.

The part positioning accuracy of a system which constrains the location of a part in the grippers is somewhat decoupled from the accuracy of part fixturing provided by the part jigs. As long as the part jigs deliver the parts within a certain range, the parts will be reoriented by the grippers, although the constraint might not be in all degrees of freedom.

Other contributing errors are not discussed here. Errors from tolerance stackup and clearance buildup are presented in References [34,189] and robot repeatability and teaching errors are discussed in Section 2.10.

2.9 Errors in the *Measurement and Removal* Method

This section discusses errors which are generated in a system using a single part position sensor. All references to an assembly procedure refer to the task described in Table 2.2a.

2.9.1 Error Sources

We model the errors in aligning a part with its mating part as being from five major sources.

$$\delta\tilde{T}_{P_{AA.A}} = \overbrace{\delta\tilde{T}^{teach}_{P_{AA.A}}}^{\text{teaching}} + \overbrace{\delta\tilde{T}_{R_{AA}}}^{\text{robot positioning}} + \overbrace{\delta\tilde{T}_{\text{fr. shift robot}}}^{\text{robot kinematics}} + \overbrace{\delta\tilde{T}_{\text{fr. shift sensor}}}^{\text{robot}-\text{sensor align.}} + \overbrace{\delta\tilde{T}_{\text{command}}}^{\text{calculated transform}} \tag{2.37}$$

Errors not being considered in this analysis are those associated with the part model inaccuracies and out of tolerance parts, non-orthogonal coordinate systems, stackup of part tolerances and clearances and certain robot positioning errors.

Table 2.3 shows the errors which occur during the calibration and teaching phase (step 0). and Table 2.4 shows the pertinent errors which occur during the task execution phase (steps 1 through 6).

2.10 Errors in a Typical Assembly Task

This section describes a hypothetical peg-in-hole assembly task which is performed using the *measurement and removal* method. Requirements for a part position sensor which can reliably perform the task in conjunction with a PUMA manipulator are calculated. Because of the relative sparsity of robot accuracy data in the literature, some of the values of the errors may not accurately correspond to actual

Step	Variable	Description
0	${}^{S}\delta\tilde{T}_{W}^{calib}$	Error due to the calibrated alignment of the sensor frame and the motion of the robot (world frame).
	$\delta\tilde{T}_{R_{AA}}^{teach}$	Error in the robot position at the assembly approach.
	$\delta\tilde{T}_{R_S}^{teach}$	Error in the robot position at the sensor.
	${}^{S}\delta\tilde{T}_{P_S}^{teach}$	Error in the location of the object by the sensor in the sensor coord. frame.
	$\delta\tilde{T}_{P_{AA,A}}^{teach}$	Error due to initial alignment of the part and the assembly. It is due predominantly to user errors.

Table 2.3: Vectors corresponding to transformation errors which occur during the calibration and teaching phase.

Step	Variable	Description
3	$\delta\tilde{T}_{R_S}$	Error due to positioning of the robot at the nominal sensing position.
4	${}^{S}\delta\tilde{T}_{P_S}$	Error in the location of the object by the sensor in the sensor coord. frame.
5	$\delta\tilde{T}_{R_{AA}}$	Error due to positioning of the robot at the nominal assembly position.
	$\delta\tilde{T}_{R_{W,WA}}$	Vector corresponding to transformation of the apparent robot world coordinate system as the robot moves from a position near the sensor to a position near the assembly. This is due primarily to inaccuracies of the robot's internal model of its kinematics.

Table 2.4: Vectors corresponding to transformation errors which occur during the task execution phase.

values for certain tasks or particular robots. Most of the numbers mentioned in the following sections are standard deviations *not* maximum values.

The task consists of the insertion of a 1.75 inch (44.5 mm) square cross-section aluminum peg into a steel hole with .004 inches (.1 mm) clearance on each side (clearance ratio is $c \approx .0025$). The hypothetical peg has small chamfers (about .02 inches or about .5 mm) and the hole is chamferless. The task corresponds to one of the demonstration tasks in Section 6.6.1.

Analysis of the Task

In order to successfully complete the task, both the non-wedging criterion, Equation (2.2), and the chamfer bound criterion, Equation (2.1), have to be satisfied. We assume that the center of rotational compliance is 10 inches (254 mm) from the tip of the peg (L_g = 10 inches) and that the displacements from the translational and rotational compliances are about the same. ($\frac{K_\theta}{K_x} \approx L_g^2$). With these assumptions, the criterion for successful task completion is

$$\left|\theta_0 + \frac{\epsilon_0}{2L_g}\right| < \frac{c}{\mu} \tag{2.38}$$

and

$$|\epsilon_0| < L_{chamfer}. \tag{2.39}$$

where ϵ_0 is the initial translational misalignment perpendicular to the insertion direction, θ_0 is the initial angular misalignment, and $L_{chamfer}$ is the size of the chamfer.

The coefficient of friction between an aluminum peg and a steel hole is about 0.3 [19]. The wedging constraint becomes

$$\left|\theta_0 + \frac{\epsilon_0}{20}\right| < \frac{.0025}{.3}.$$

Thus for reasonable misalignment errors the wedging criterion is dominated by the rotational offsets ($|\theta_0| \gg \left|\frac{\epsilon_0}{20}\right|$) and the chamfer crossing failure criterion depends on the displacement offsets; thus, the errors decouple and may be investigated separately.

The maximum allowable displacement error is [from Equation (2.1)] $\epsilon_0 = \pm L_{chamfer} = \pm.02$ inches (± .5 mm). The maximum allowable rotational error is [from Equation (2.2)] $\theta_0 = \pm\frac{c}{\mu} = \pm 8.5$ milliradians (.5 degrees). If 98.8 percent of the assembly trials (5 standard deviations) are to be successful, the allowable standard deviations in errors are

$$\sigma_{\epsilon_0} = \frac{2(.02)}{5} = .008 \text{ inches } (.2 \text{ mm})$$

$$\sigma_{\theta_0} = \frac{2(8.5)}{5} = 3.4 \text{ milliradians } (.2 \text{ degrees})$$

Teaching Errors

Teaching errors discussed here are those that the user is directly accountable for. With a single sensor system, the alignment of the part with its mating part in the assembly, $\delta \tilde{T}^{teach}_{P_{AA.A}}$, is the sole error source. All other errors during teaching are accounted for in the commanded transform error. It is assumed that the human operator can specify a position for a low clearance ratio mating operation to within a standard deviation of .002 inches (.05 mm) in translation and .1 degrees (.0017 radians) in rotation. The values of these numbers will vary depending on what, if any, measuring tools are used to aid the alignment.

Robot Positioning Error

Only the error in positioning at the assembly approach position is included here; the robot positioning errors at other locations in the workcell are accounted for in subsequent sections.

We assume that this error is equal to the robot repeatability (other local inaccuracies of the manipulator are taken into account in the kinematics error). Repeatability is the error associated with the robot moving to a position associated with a certain set of joint angles. It is usually measured by having the robot move to random positions in between measurements at the position of interest. Repeatability of the PUMA robot was investigated by Lozinski [117] and Whitney, Lozinski and Rourke [203]. Maximum values rather than statistical data was presented; therefore, the repeatability standard deviation is estimated to be $\frac{1}{4}$ of the reported value, $\frac{1}{4}.004 = .001$ inches (.025 mm). No information was found on orientation repeatability in the literature.

The repeatability of a PUMA robot was measured with the prototype sensor. The errors were of the same order as the sensitivity of the sensor so accurate readings could not be made. The readings obtained may, however, be used as a maximum bound for the standard deviation repeatability of the manipulator; they were .001 inches (.025 mm) translation and .06 degrees (.001 radians) in rotation (see Section 6.5).

Robot Kinematics Error

A robot does not move precisely in its workspace because of an imprecise model of its link and joint parameters, finite position encoder accuracies, structural deformations, transmission errors, etc (see [117,203] for a discussion). We assume that errors from all of these sources may be modeled by an error in the location of the world frame. A manipulator will execute a given transformation with respect to a slightly different world frame as certain conditions change. If it were to execute the transformation with respect to a single world frame, it would have infinite accuracy. This error taken over a small region and combined with the robot repeatability is a measure of the local accuracy of the robot.

Here we consider errors which are generated from inaccuracies in the location of the world frame and deviations from a nominal taught path. The orientation error is given by Equation (2.35) and the translation error by Equation (2.36). We assume that the location of the part (measured location) from the nominal location (taught location) is described by a symmetric Gaussian of width 1.5 degrees (.024 radians) in rotation, and the PDF for the error in the robot's world coordinate frame has width .05 inches (1.3 mm) in translation and .5 degrees (.008 radians) in rotation. In addition, we assume that the starting coordinate of the robot is at position $\vec{d} = [12 \;\; 12 \;\; -12]^T$ inches ($[305 \;\; 305 \;\; -305]^T$ mm). From Equations (2.35) and (2.36) the standard deviation of the final error in each direction is

$$\sigma_{\alpha_{kinematics}} \approx \sqrt{2(.024)^2(.008)^2} = .0005 \text{ radians} \;\; (.015 \text{ degrees})$$

in rotation and

$$\sigma_{\epsilon_{kinematics}} \approx \sqrt{.024^2[2(.05)^2 + 4(12)^2(.008)^2]} = .005 \text{ inches} \;\; (.12 \text{ mm})$$

in displacement.

Sensor-Robot Coordinate Alignment Error

Errors in robot motion due to a coordinate frame misalignment with the sensor may be modeled with the same equations as the robot kinematics error. In this case, the error in the world frame is due to the misalignment between the sensor and the robot frames with the error specified near the sensed part feature ($\vec{d} \approx 0$). The error is assumed to be .05 inches (1.3 mm) in displacement and 2 degrees (.032 radians) rotation.

$$\sigma_{\alpha_{sensor-robot}} \approx \sqrt{2(.024)^2(.032)^2} = .001 \text{ radians } \ (.06 \ \text{ degrees})$$

$$\sigma_{\epsilon_{sensor-robot}} \approx \sqrt{2(.024)^2(.05)^2} = .0017 \text{ inches } \ (.043 \text{ mm})$$

Thus, the total error is fairly small even with a relatively large sensor-robot misalignment.

Errors in the Commanded Transformation

The errors not yet accounted for include: the sensor measurement error (${}^S\delta\tilde{T}_{P_S}$), the sensor measurement error during teaching (${}^S\delta\tilde{T}_{P_S}^{teach}$), the robot repeatability in positioning the part at the sensor ($\delta\tilde{T}_{R_S}$), and the robot repeatability in positioning the part at the sensor during teaching ($\delta\tilde{T}_{R_S}^{teach}$). The errors which occur during teaching become embedded in the nominal transform which moves the part from the sensor position to the assembly approach position. After a number of executions of the assembly procedure, the commanded transform may be manually corrected until most of these embedded repeatable errors are nullified; thus, they were not included in the error budget. The only remaining errors are ${}^S\delta\tilde{T}_{P_S}$ and $\delta\tilde{T}_{R_S}$. The robot repeatability is assumed to be .001 inches (.003 mm) in translation and .06 degrees (.001 radians) in rotation; thus, the error due to the calculated command transformation is

$$\sigma_{\epsilon_{command}} = \sqrt{{}^S\sigma^2_{\epsilon_{P_S}} + .001^2}$$

$$\sigma_{\alpha_{command}} = \sqrt{{}^S\sigma^2_{\alpha_{P_S}} + .06^2}$$

Total Errors in an Example Assembly System

Quantitative values for the five error sources in Equation (2.37) are

$$\sigma_{\epsilon_P} = .008 = \sqrt{.002^2 + .001^2 + .005^2 + .0017^2 + .001^2 + {}^S\sigma^2_{\epsilon_{P_S}}} \text{ inches} \qquad (2.40)$$

in translation and

$$\sigma_{\alpha_P} = .2 = \sqrt{.1^2 + .06^2 + .015^2 + .06^2 + .06^2 + {}^S\sigma^2_{\alpha_{P_S}}} \text{ degrees} \qquad (2.41)$$

in rotation. Thus, the allowable standard deviations in measurement accuracy for the sensor for the hypothetical peg-in-hole task are

$${}^{S}\sigma_{\epsilon_{P_S}} = .0055 \text{ inches } \ (.14 \text{ mm})$$

$${}^{S}\sigma_{\alpha_{P_S}} = .14 \text{ degrees } \ (.0025 \text{ radians})$$

Part Position Sensing for Assembly

Chapter 3

3.1 Literature Review on Vision Based Part Sensing

3.1.1 Ranging Systems

Identification and three-dimensional position measurement of objects require a sensing system which can detect points or features on the object's surface. Contact or non-contact sensing techniques may be used to acquire surface position data. Non-contact systems are generally faster, more versatile and higher resolution than contact systems. Vision based systems are usually highly flexible and have high resolution, but are sometimes slow. Jarvis [99] presents an overview of various ranging techniques including light stripe systems, texture gradients, range from focusing, stereo disparity, range from motion, moire fringe contours, single spot triangulation, and time of flight measurements. Joseph and Hansel [23] also give an overview, but it is predominantly a concise version of Jarvis's article. Benton and Scarborogh [23] describe some commercially available systems. Techniques for obtaining depth information not cited in the aforementioned literature include an optical proximity sensor [100], projection of regular patterns [184], focusing a ring pattern [102], and a technique which servos the light source on a positionable sensor [12]. A discussion of some systems which use the light stripe ranging technique follows.

The "light stripe" technique for obtaining three dimensional measurements of points on surfaces of objects uses a planar light source projected across the scene. The light source is usually either a white light projected through a slit or a thin laser dispersed in one direction through a cylindrical lens. The scene is usually sensed with a video camera. Three-dimensional coordinates may be calculated for each illuminated point in the image (see Section 4.2.2 for a more detailed description). Shirai and Suwa [175] scanned a light stripe across a scene containing polyhedral objects. They segmented the planar surfaces by detecting discontinuities in image stripe slope and spacing. Planes were then fitted to the points from lines in a group. Agin and Binford [5] fit data generated from multiple images of a scanned light stripe to generalized cylinders. Their technique only worked well on parts of objects which were close in structure to a generalized cylinder. Popplestone, Brown, Ambler and Crawford [154] were able to construct models of objects composed of planar and cylindrical surfaces from light stripe data. They clustered segments of the light stripes and attempted to fit planes or cylinders to each cluster.

A sensor system developed at the National Bureau of Standards [7] used two parallel light planes and a point source of light. Two images are taken. The first, using the planar sources, is used to get range, pitch and yaw information. The second, using just the point source, obtains position information perpendicular to the optical axis and roll information. Because only two planes were used in the NBS sensor, there was usually no confusion about which source a line in the image corresponded to. In general, however, multiple planar sources can create stripe to source correspondence problems [162]. By using multiple cameras, Echigo and Yachida [56] solved the multiple stripe identification problem.

Cain [37] uses curve matching to inspect a motor end bell and a plastic bottle from light stripe data. He is able to filter out spurious reflections by checking that the direction of the ray from the source to the illuminated points in adjacent line segments are consistent. Other references which use a light stripe ranging system for object recognition or inspection include [4,23,129,130,155,156,183,195,198,209]. This type of ranging system has also been extensively used for robotic welding (see [3] for a reference list).

Accuracies in locating three dimensional features using these light stripe systems were usually not presented. Since little attention appeared to be given to precise calibration, it is likely that the accuracies of these techniques were poor with respect to the requirements for a vision-driven precision-assembly task (Section 3.4 discusses these requirements).

3.1.2 Model Based Object Recognition and Position Determination

Overviews of model based vision systems for identification and location of two and three dimensional parts in a scene are given in [24,44,45]. A model based system assumes *a priori* models of objects potentially in a scene. The goal is usually to identify the objects and determine their position through matching with the models.

Ikeuchi et al. [95] used a photometric stereo technique [206,93] to determine the orientation of objects of known shapes and known surface properties stacked in a pile. A needle map (surface normals plotted over the image) of the scene was formed and used to segment the scene into regions corresponding to different objects. The Extended Gaussian Image (EGI) [17,93,177] was then used to determine the orientation of a selected object. An EGI is essentially a mapping of all the surface normals of an object onto the surface of a sphere. After determining the object's orientation, a grasp point for the object was chosen and a manipulator was instructed to pick the object out of the pile. Brou [35] also used the EGI to determine the pose of objects whose surface normals were calculated from data points generated from a laser ranging system.

Oshima and Shirai [145,173,174] use a region growing algorithm on range data to construct planar and quadric surface patches corresponding to surfaces of objects in the scene. A kernel region from the scene is used to search possible models for a correspondence (data driven search). Once found, regions surrounding the kernel region in the candidate models are used to search through the scene for additional matches (model driven search) until enough regions of a particular object are found. This procedure is repeated for each object in the scene. Oshima et al. also report using two other techniques: a photometric stereo-EGI technique for certain shaped parts , and a polarimetric technique [106] for somewhat specular objects.

Faugeras et al. at INRIA [58,59,60] have developed a system which represents, recognizes, and finds the position of three-dimensional objects from range data. Objects are modeled by points, lines, planes, and quadric surfaces. They use a hypothesize-and-test algorithm for determining the relative position of a sensed object to a model. A rigidity constraint is used to help determine an initial hypothesis. They have developed techniques for finding the best-fit rotation and translation which, although computationally efficient, fail to take into account the relative measurement accuracy of each of the scene normal vectors.

Horaud and Bolles [91,28] have developed a three-dimensional feature based system to recognize parts within a jumbled pile. The parts are modeled with cylindrical and planar surfaces and a list of features is associated with each part type. Light stripe range data is used to obtain points on the surfaces of the parts. Edges between surfaces are found and classified as lines or arcs. An edge is matched to possible model features then additional features in the image are used to guide a tree search to converge to the proper interpretation. Tomita and Kanade [193] have developed a similar feature based approach to the matching problem. They find circular and straight surface bounding edges which are used as the matching features. An initial match is found by essentially an exhaustive search. They suggest that after a number of feature matches have been found, the hypothesized transformation may be more accurately determined by performing a least squares minimization of the errors between all the matched features.

Lozano-Perez and Grimson [76,77,78,79] use local constraints on geometric features to prune an interpretation tree of possible three-dimensional object configurations. Earlier work described a similar technique for determining the position of an object in a plane from sparse tactile data [63]. These techniques need only sparse scene data; thus, the time needed to acquire three dimensional data from a ranging system, which can sometimes take tens of minutes, may be substantially reduced. Their technique is robust to partial occultation and is a possible solution to the bin picking problem. This technique is very good at finding gross object positions, but like most of the other systems described in the literature, does not always produce accurate location information. This sparse range technique may, however, be used to determine the pose of an object before it is grasped by a manipulator. Once grasped, the part may be repositioned so that a precise position measurement may be taken by a sensor such as the one described in this thesis.

The following is a brief description of some other techniques which have been used to determine the three-dimensional position of objects in a scene. Goad [67] describes a technique which matches discontinuities in two-dimensional image intensity to features in a part model. A matching hypothesis is formulated and checked with multiple images from cameras positioned around the part. When a certain number of edges agree in all views, the hypothesis is assumed to be valid.

Herman [86] generates three dimensional models from light stripe range data by finding discontinuities in the data (edges) using Hough transforms, then identifies vertices and faces along with the edges to complete the model. Hébert and Kanade [82] also model a part by its edges (they call this representation a 3-D profile). They discretize the space of possible orientations of the object and precompute occluding boundaries of a hypothesized image of the object in each of the orientations. In the recognition phase, they attempt to match the precomputed occluding boundaries to those in the actual image. The accuracy of the technique is highly dependent on the fineness of the tessellation of the orientation space. Hough transforms have been used in a non-feature based approach [18,111,127]. Edge curves, and planar and quadric faces of objects in a scene may be found directly from the data.

3.1.3 Vision-Sensor-Driven Assembly

Sensor based assembly literature deals almost exclusively with force control and active and passive compliance techniques (see [202] for an overview); very little attention has been given to sensing in the alignment phase of an assembly task. Shirai and Inoue [172] used a video camera mounted on the manipulator to monitor a peg-in-hole insertion. Part alignment corrections were made as the assembly progressed. Only two dimensional data was used and the bandwidth of the system was very slow due to the image processing step. Inoue and Inaba [97] describe a "hand-eye" system based on binocular stereo which can perform manipulations with a length of rope. A commercial automated electronic component assembly system (Automatix Multisert system) uses vision-driven assembly for both surface mount and through hole components. Benton proposed using a light stripe system to monitor and correct for errors in the the assembly of microswitch parts [23]. Park [23] discusses this technique and some problems encountered, including sensing positions of specular parts. Rutkowski and Benton [23] report on the algorithm used to determine the pose. They use an iterative algorithm which transforms imaginary data points in a part model until they align with the sensed data in the laser stripe scene. Theoretical accuracy of part alignment for their system was .007 inches (.2 mm) and experimental accuracy under ideal conditions was .015 inches (.4 mm). Experiments on real parts showed errors as much as .033 inches (.8 mm). Test procedures and statistical results were not given.

3.2 Locating Objects from Range Data

In this section, the question of which type of model fitting algorithm should be used on raw range data is addressed. One method tries to fit the range data to one of a number of previous observations of the part while another method tries to fit an *a priori* mathematical model to the data.

3.2.1 Introduction

The precision of two techniques for finding the orientation of objects using range data is investigated. The first method, a *fitted boundary interpolation* technique, uses range sensor data recorded during a learning phase from many sensor scans of a sample part at various precisely controlled positions about a nominal position. This data is later used as the source for an interpolation routine which estimates the exact position of a similar part based on data from a single scan. In the second method, a *feature locating* technique, an accurate geometric model of the part features of interest is used in conjunction with sensed data to calculate the orientation and position of the part. Results from two-dimensional studies using these techniques produce insight on the performance of the two algorithms in locating actual three dimensional parts. Performance is evaluated as a function of the number of data points, and the shape and the orientation of the object producing the data.

3.2.2 Method of Evaluation

The Sensor Model

A generalized three-dimensional range sensor is modeled in two dimensions by a one dimensional camera array sensing contours (x, y coordinates of "visible" boundaries) of a two dimensional object (see Figure 3.1). The effect of discretization due to the geometry of the sensing array and the finite resolution in range information is studied. The width of the image and the precision of the range data are both assumed to be represented by eight bit numbers (256 × 256 array).

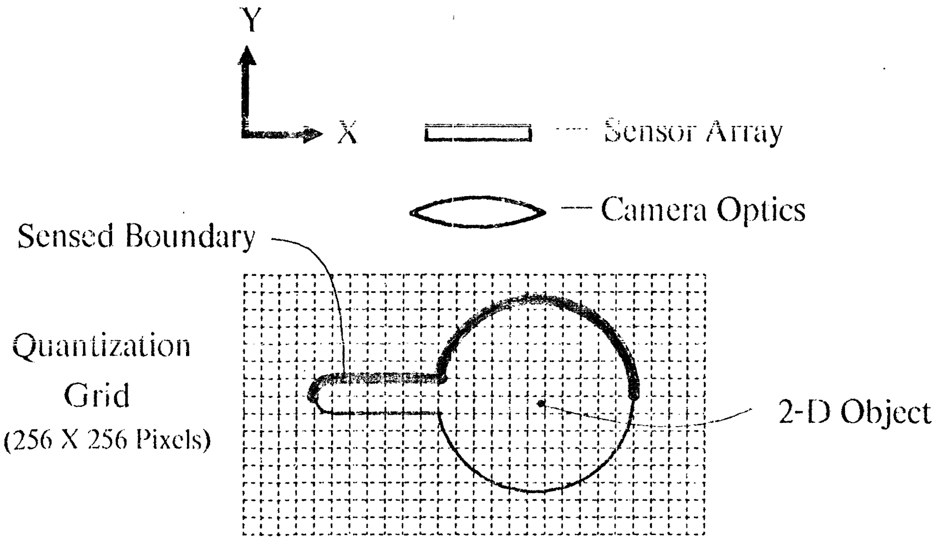


Figure 3.1: Two dimensional model of a ranging sensor and object

The Fitted Boundary Interpolation Technique

In this method, an interpolation table is created based on data from scans of the part in a series of known orientations. A curve or surface is fit to the data for each orientation and parameters describing each curve or surface are stored in a table. The orientation of a sensed object is found by using newly fitted curve parameters as an index into the table and interpolating the object's orientation from the coefficients. For the present study, a straight line is fitted to the surface contour data and the two element lagrangian interpolation technique (linear two point interpolation) is used to calculate the orientation of the object (usually different from the orientation of the fitted line) from values stored in the table. The dependency of the algorithm on the density of the entries in the table is shown in Figure 3.2. As the density of the table entries increases, the algorithm becomes more accurate. When the spacing between orientation entries is less than about 10 minutes, (this corresponds to the outer edge of the image moving about 0.18 pixels from one entry to the next), the algorithm does not become significantly more accurate. At this "saturation" point, fitted slopes of almost all permissible pan images have one or more corresponding entries in the table. When the fitted boundary parameters correspond to multiple entries in the interpolation table, the mean of the table values for the identical entries is utilized.

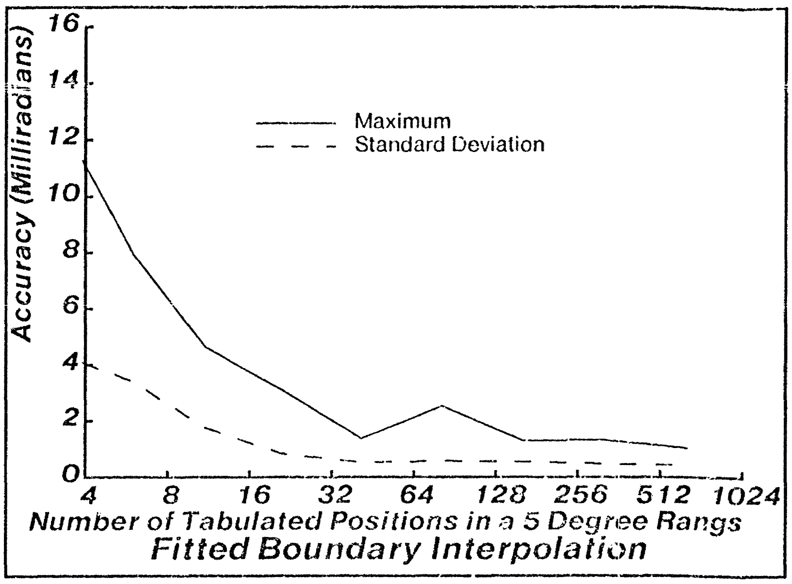


Figure 3.2: Dependency of the fitted boundary interpolation technique on table density for the pan image oriented between 5 and 10 degrees.

The Feature Locating Technique

In two-dimensions, relatively straight-line features are identified and used to perform matching between a model and the sensed data. For example, for the simulated image of a pan shown in Figure 3.3a, the straight section of the handle may be used to determine the orientation of the entire image. We will use the results of Section 4.3 which present the errors in least squares line fitting to evaluate this technique in the two dimensional case.

3.2.3 Studies

The magnitude of the errors from a sensor with discrete elements is sensitive to the exact position and orientation of an object's image within the pixel array. Small changes in image position can have a large effect on the error. Errors were estimated by calculating sample standard deviations of randomly oriented images within some orientation range (either a 2.5 or 5 degree range). Sample statistics from 30 trials in each range were calculated and used as bases for comparisons.

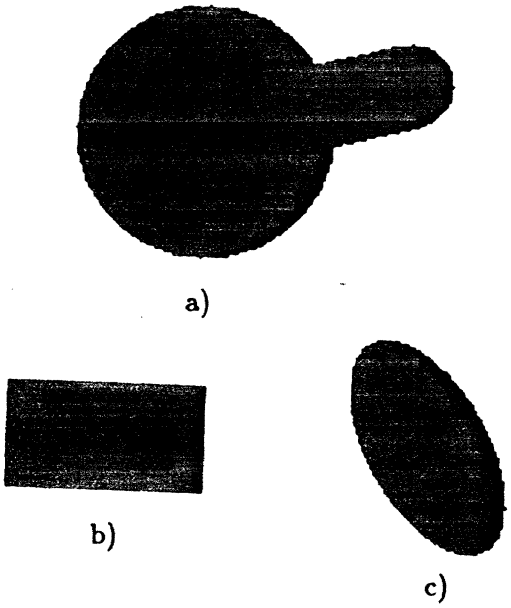


Figure 3.3: Two dimensional images used to test the accuracy of the part locating algorithms. a) Pan image. b) Rectangle image. c) Ellipse image

Sensitivity to Image Size

In evaluating the fitted boundary technique, measurements were interpolated from a table consisting of 6 equally spaced entries from 5 to 10 degrees (1 degree increments). The pan image was used in the evaluation. After the interpolation table was constructed, the image was oriented precisely in the range from 5 to 10 degrees. Contour data was generated from the sensor model and the resulting discrete image was evaluated to get an estimate of part position. This estimate was then compared to the actual part position and the difference was recorded. The length was varied (the aspect ratio is kept constant) and the errors in measured orientation recorded. Errors in orientation as a function of image length are shown in Figure 3.4.

The feature locating technique was evaluated as follows. A randomly oriented image (within a range of 5 to 10 degrees) of a straight line portion of an object was located by least squares line fitting data from the sensor model. Differences in orientation between the measured and actual orientation were recorded for different length edge images. Results are shown in Figure 4.13 in Section 4.3.

Although the results (standard deviations of errors in orientation) from the fitted boundary interpolation technique appear to be quite inferior to the results

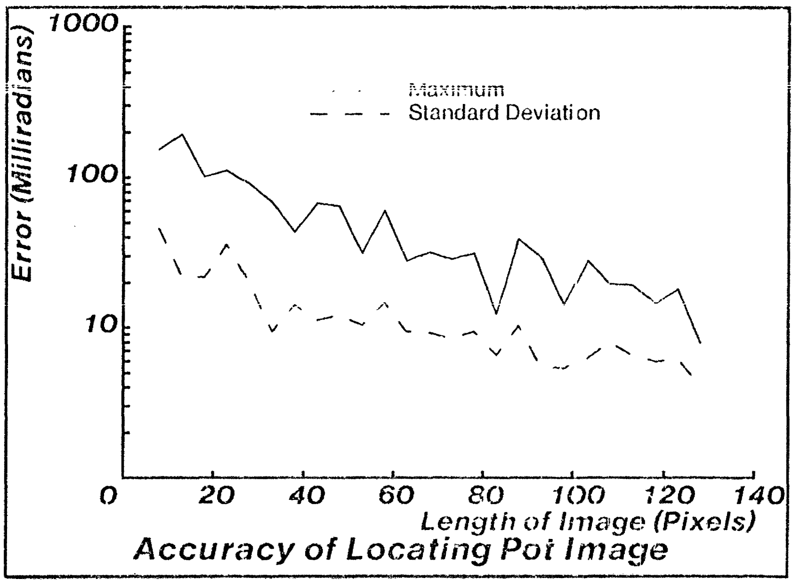


Figure 3.4: Dependence of the fitted boundary interpolation algorithm on the length of an image of a pan whose orientation is between 5 and 10 degrees

of the feature locating technique, they are extremely sensitive to the interpolation table density (see Figure 3.2). In the limit of a fully saturated table, the fitted boundary interpolation table has explicit values for all feasible discrete images and gives results of equal or greater accurate than the feature locating approach. Variables which determine the accuracy of the fitted boundary interpolation technique with sub-saturation table densities are discussed in subsequent sections.

Image Orientation

The accuracy of the fitted boundary interpolation algorithm as a function of image orientation was investigated. In this test statistics from 30 trials in each 5 degree range from 0 to 65 degrees were collected. Orientation tables with one degree increments for both the pan image and the rectangle image were used. Images of the pan and subsequently of the rectangle were presented to the sensor within the working range of the table. The results are shown in Figures 3.5, and 3.6. Errors in orientation for the pan image are larger and vary more than those for the rectangle image. The decrease in accuracy with increasing angle as seen in the

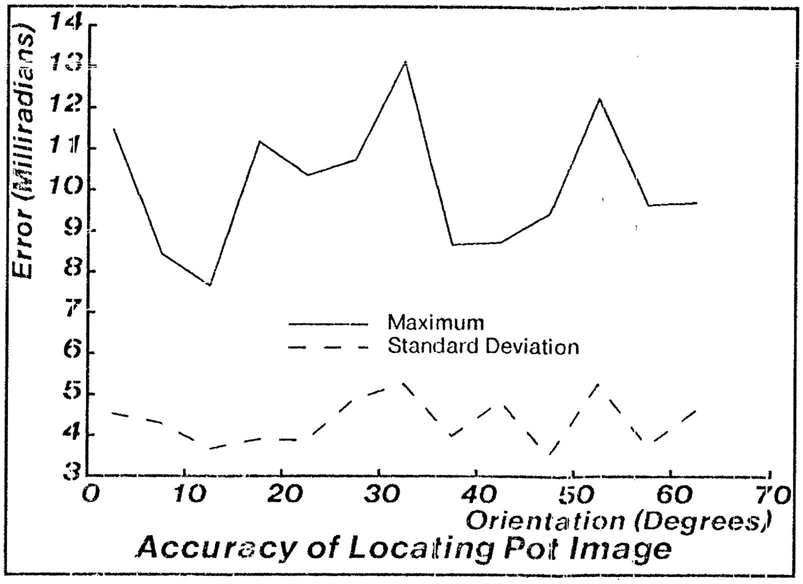


Figure 3.5: Dependency of the fitted boundary interpolation technique with image orientation for an image of a pan

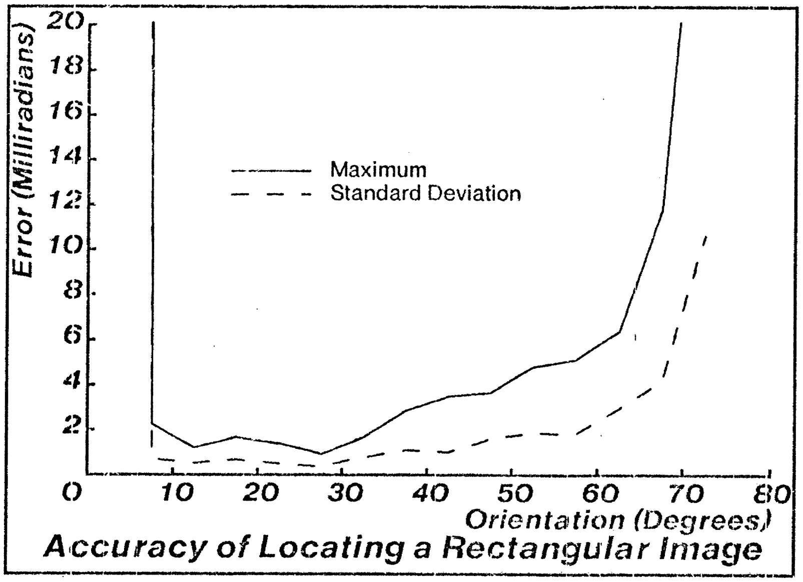


Figure 3.6: Dependency of the fitted boundary interpolation technique with image orientation for an image of a rectangle

rectangle is due to the decrease in projected length of the image of the rectangle and to an increase in the number of discontinuities in the interpolation table as described later.

The accuracy of the feature locating technique as a function of feature orientation is the same as the accuracy of the linear least squares technique (see Figure 4.10 in Section 4.3).

The rectangle interpolation table with 1 degree increments was sufficiently fine to produce errors which were comparable to those of the feature locating (least squares line fitting) technique while the pan table with 1 degree increments produced errors 5 to 10 times greater than the feature locating technique.

Image Shapes

The errors in locating an image using the fitted boundary interpolation technique can change more than an order of magnitude depending upon the shape of the object. The closer the shape of the object is to a straight line, the better the performance of the algorithm. This conclusion is supported by the fact that the method more accurately locates the rectangle than it does the pan (Figures 3.5 and 3.6 and Section 3.2.3). Rectangles and ellipses with large aspect ratios are more accurately located than similarly shaped images with aspect ratios approaching 1 (Figures 3.7 and 3.8). This shape dependence is due to the number and magnitude of the discontinuities in the orientation tables. Comparing the number of discontinuities in the table generated for the pan image (aspect ratio ≈ 2 – Figure 3.9) to the tables for ellipses and rectangles of various aspect ratios (Figures 3.10 and 3.11), we may conclude that the longer and straighter the object, the higher the accuracy of the fitted boundary technique. Only the first part of the interpolation table (where the table was single valued) was used. Table interpolation of orientations above about 65 degrees were not performed.

The discontinuities in the interpolation table are an artifact of the discretization of the image. At certain positions, small rotations of an object may change only a few pixels states; alternatively, a large number of pixels may change state. If subsequent small rotations of the object (in the same direction) during the learning phase of the fitted boundary technique generate very few pixel state changes then a large number of pixel state changes, a discontinuity in the interpolation table will result. Since the pixel state changes may happen simultaneously, a very fine

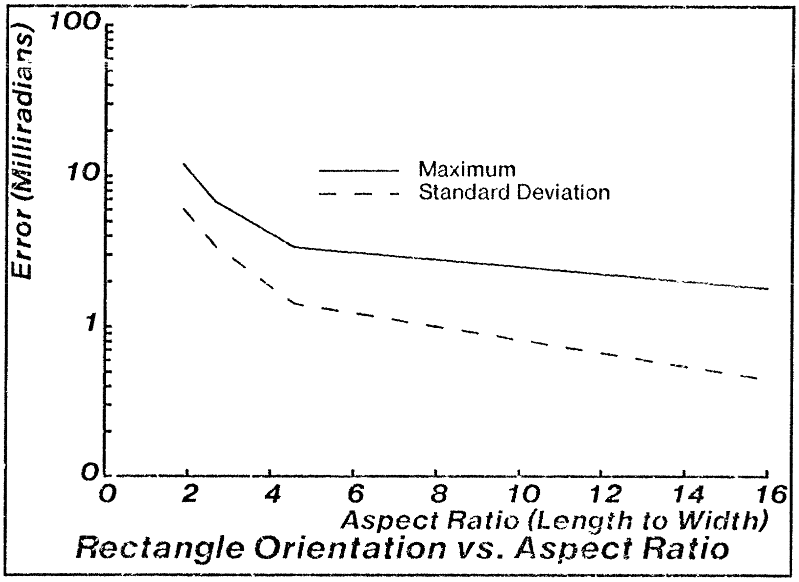


Figure 3.7: Dependence of the fitted boundary interpolation technique on the aspect ratio of a rectangular image

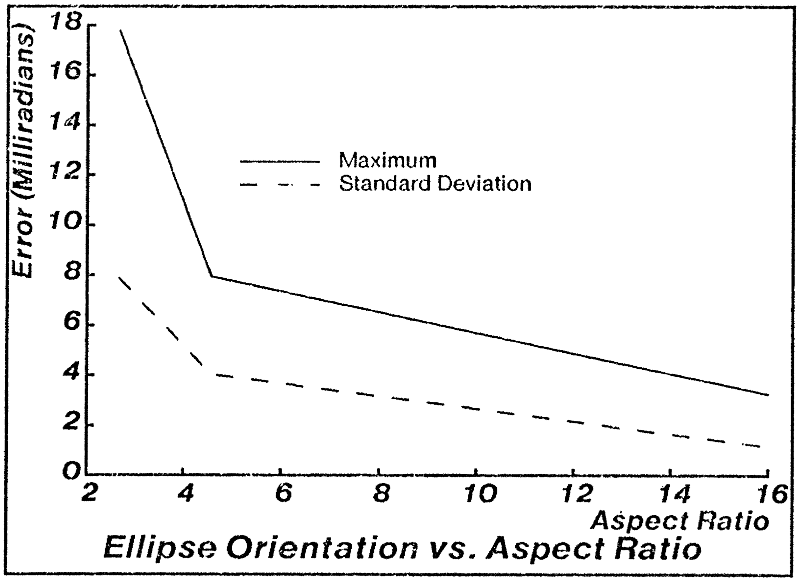
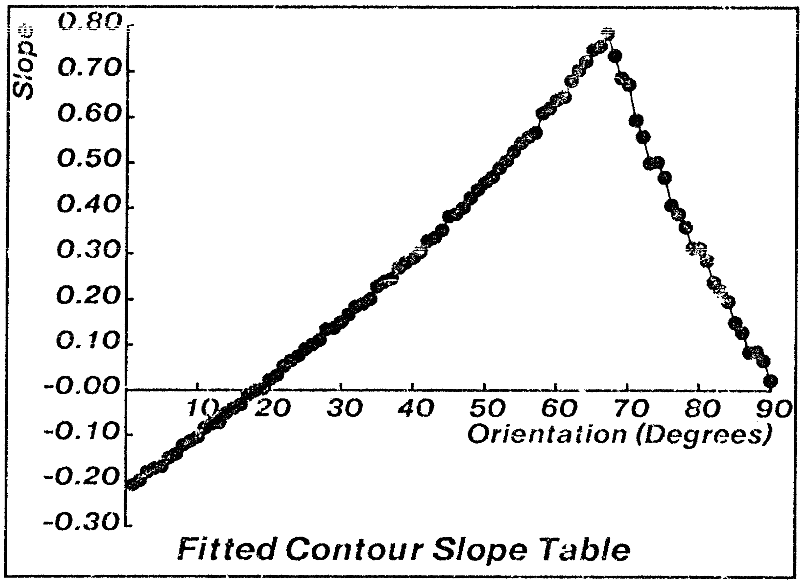
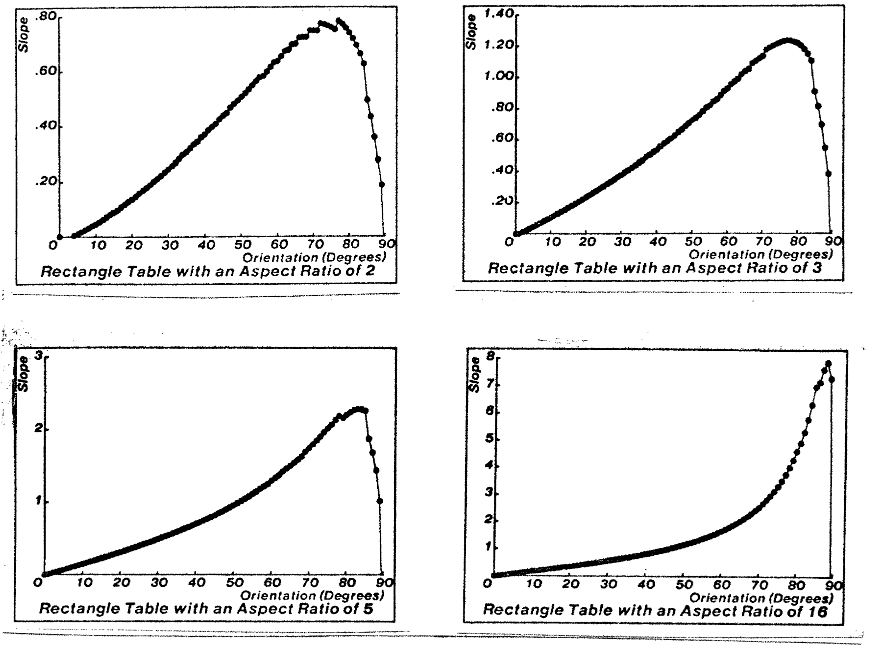


Figure 3.8: Dependence of the fitted boundary interpolation technique on the aspect ratio of an elliptical image



Fitted Contour Slope Table

Figure 3.9: Pan image table entries for the fitted boundary interpolation technique



Rectangle Table with an Aspect Ratio of 2

Rectangle Table with an Aspect Ratio of 3

Rectangle Table with an Aspect Ratio of 5

Rectangle Table with an Aspect Ratio of 16

Figure 3.10: Tables for rectangles of various aspect ratios

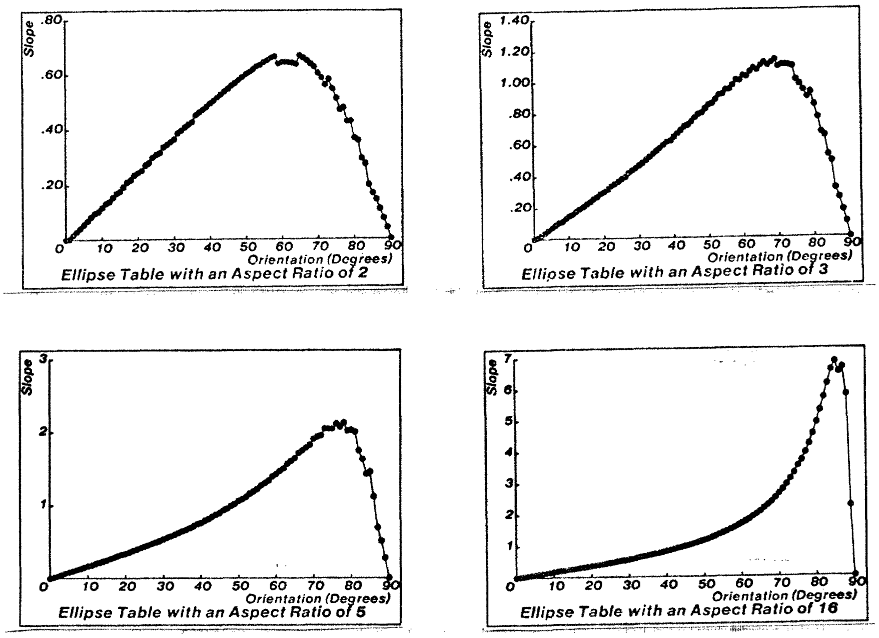


Figure 3.11: Tables for ellipses of various aspect ratios

interpolation table may also exhibit many discontinuities. Figure 3.12 shows the quantized outline of an ellipse. A small rotation, $\delta\alpha$, of the ellipse can bring about a relatively large rotation in the line fitted to the upper boundary of the ellipse due to the change of state of a number of the pixels on the right and left boundaries. This change in the fitted line orientation results in a discontinuity in the interpolation table.

Since the feature locating technique studied herein is only capable of locating straight line features, it will not be discussed in this section.

3.2.4 Conclusions

- Accuracies of the fitted boundary interpolation technique approach those of the feature locating technique if the table density approaches saturation. Some shapes may be located accurately without using a fully saturated table. These shapes have relatively smooth (few discontinuities) interpolation tables.

- Both techniques perform best when the "longest" side of the object is fac-

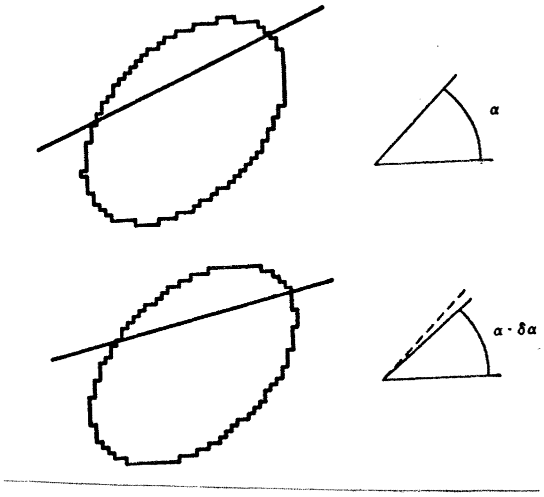


Figure 3.12: A small rotation, $\delta\alpha$, in the discretized ellipse can bring about a relatively large rotation of a line fitted to the upper boundary of the ellipse.

ing the sensor so as to reduce image foreshortening. The feature locating technique requires objects to have certain features while the fitted boundary interpolation technique works best with relatively thin, straight objects.

- Extending the number of required elements in an interpolation table for a six degree of freedom boundary interpolation technique suggests an extremely large interpolation table and lengthy searching algorithms which may not be appropriate for a real time industrial environment.

- Although we have assumed no *a priori* model for the fitted boundary interpolation technique, the location of the example part in the learning phase must be positionally referenced to a base coordinate frame. This requires defining some local coordinate frame on the part from which the sensed boundary is referenced. Thus, the technique cannot be used without constructing some sort of *a priori* model; although, the model can be quite simple. The model needed for the feature locating technique can also be simple since only the features which are to be sensed need to be modeled.

- Both object locating techniques have limitations. The fitted boundary interpolation technique cannot handle objects with aspect ratios near one and the feature locating technique requires an object to have easy to sense and easy to model features. As implemented the fitted boundary interpolation technique will only work about a nominal orientation where the coefficients of the fitted contour curves are single valued.

- The two-dimensional feature locating approach may be applied to objects in three dimensions and still remain relatively fast (perhaps a few tenths of a second) and relatively accurate; however, it is currently limited to finding only fairly simple geometric features.

3.3 Assembly Systems Which Use a Part Position Sensor

Programmable assembly systems such as the one described in Chapter 5 often require special tooling. Trading jigs, fixtures, end effectors, and special part pallets into and out of the assembly environment produces a less flexible (less able to handle different products, part shapes, and assembly operations) and less efficient system. Instead of precisely fixturing parts, a sensing technique may be used to localize features of a part. In addition to making the assembly cell more flexible, a part position sensor can make the assembly operations more reliable. Factors such as out-of-tolerance parts, slightly out-of-position parts, burrs, and worn fixtures contribute to the "error budget" for an assembly operation. Most of these error sources can be eliminated if a part position sensor is used to sense the mating features of parts just prior to their assembly.

An assembly system which uses part position sensing in lieu of precise fixtures is shown in Figure 3.13. The system consists of a feeding station which separates and roughly orients the parts, two part position sensors, a mechanical manipulator, a series of assembly nests to hold the base parts of assemblies, and "universal" grippers which can firmly fixture parts of many shapes and sizes. One of the sensors is positioned under a transparent stage at the end of the feed track. In an assembly operation, a part is fed to the stage in an approximate orientation and the manipulator grasps it. As the manipulator lifts the part off the stage, the part position sensor takes a reading. It need not fully process the reading at this time. The manipulator then approaches the assembly. By the time the robot is ready to insert the part, the sensor has processed the measurement and the manipulator is instructed to reorient the part so that it is aligned with the mating part in the

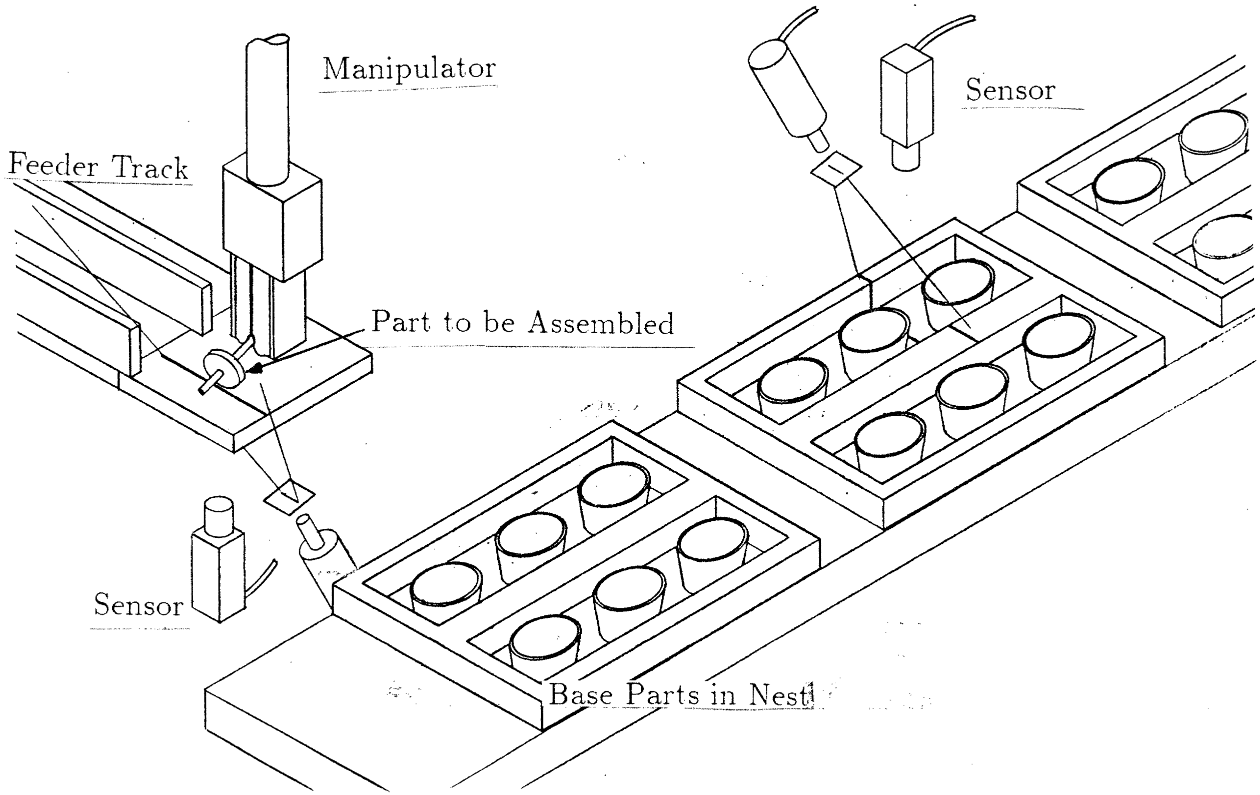


Figure 3.13: An assembly system which uses part position sensing needs no specialized fixtures. Parts are taken from a feed track by the manipulator, sensed, repositioned, then assembled. A part position sensor upline of the assembly station determines and records the position of the base parts of the assembly.

assembly nest. A second part position sensor is located upstream of the assembly station and locates the position of the mating parts in the assembly nest before they enter the assembly station.

Universal Assembly Jigs

With a part position sensor, there is no requirement that the end effector and assembly jig geometrically constrain the parts. These elements of the system need only stably (no slipping) grasp the parts. Clamping mechanisms with resilient, high friction surfaces can adequately constrain a large variety of parts.

The elimination of gripper and fixture changing during an assembly operation saves time and costs in assembly operations. An automobile alternator assembly system developed at the Draper Laboratory spent about 1/3 of its cycle time performing tool changing operations [43]. Time spent performing any non-assembly tasks means a lower throughput and, thus, a more expensive system.

Uses for a Part Position Sensor in an Automated Factory

Some uses for a part-feature-based sensing system operating in an automated manufacturing environment are

- Measure and feedback surface positions during machining operations.
- Determine manipulator endpoint positions for a servo position controller.
- Measure part positions during assembly operations.
- Inspect part features.
- Verify proper part positions after assembly operations.
- Provide part orientation information for feeding systems.
- Sense absolute end effector positions for robot calibration.

This thesis will deal mostly with the part position sensor as used in an assembly environment.

3.4 Sensor Design Requirements

In an assembly environment, a part position sensor is used to accurately locate mating part features. The sensor design goals were split into two groups; desirable attributes for the prototype sensor, and additional attributes for a commercial sensor.

Prototype Sensor Goals

- Measure to within .0055 inches (.14 mm) in translation and .14 degrees (.0025 radians) in orientation (see Section 2.10).
- Measure polyhedral part features.
- Complete measurements in less than 5 sec.
- Measure mating features if possible.
- Have a large enough sensing area to allow for initial misalignments of about ±.12 inches (±3 mm) and ±5 degrees (±.01 radians).

Commercial Sensor Design Goals

- Measure commonly found part features.
- Extendable to non-standard features.
- Complete measurements in less than 1 sec.
- Requires little or no manual intervention for sensing different parts.
- Inexpensive.
- Relatively small and light.
- Easy to calibrate.
- Safe.
- Reliable.
- Works well in an industrial environment.

3.5 Choosing a Part Position Sensing Technique

A good design for a part position sensing system will be a system which can precisely digitize and represent geometric information from the three dimensional world in a digital computer. Vision based technologies currently used to extract three dimensional information about objects include photometric stereo, binocular stereo, time of flight measurements, depth cues from two dimensional images, depth from motion and triangulation techniques (see Section 3.1 for an overview). Our interest is in locating the position and orientation of objects for automated industrial assembly tasks; thus, high accuracies and short computation times are essential. Initially a high precision stereo system which used two cameras viewing an object onto which a random texture has been projected [135] was considered; however, the processing speed for such a system was too long.

A light stripe ranging system uses a plane of light projected across a scene as the sole illumination source. Only a few points in the visual field of the camera are illuminated and consequently only a potentially small amount of data need be processed to obtain three dimensional data. Since there is an isomorphic mapping between points in the image and points on three-space, geometric computations are relatively fast. The technique used to locate objects from sparse light stripe data is presented in Chapter 4. Some other techniques use sparse range data for locating objects [76,77,78,79], but these generally need more information than the proposed light stripe system, are still too slow for most industrial tasks, and are not designed to locate objects with the required precision.

The literature review, Section 3.1, produced insufficient accuracy data to determine whether or not a light-stripe-based vision system could be used as a high precision part position sensor. A simulation study was undertaken to determine whether a carefully calibrated system could locate objects to a high enough accuracy to be used in an automated assembly environment.

3.5.1 Predicted Sensor Performance

Prior to deciding to built a prototype based on the light stripe technique, a number of computer simulations were performed to predict the accuracy available with off-the-shelf hardware. The simulations modeled a light plane source intersecting a right corner feature. The line segments generated from the intersection were

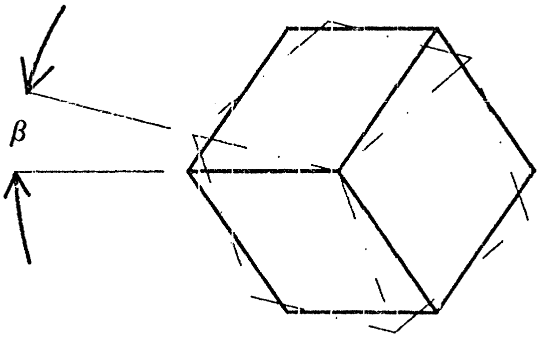


Figure 3.14: Degree of rotational freedom for simulated block tests.

perspectively transformed, sampled in a two-dimensional grid then used to locate the feature using the technique described in Section 4.7.2. The discrete grid models the picture elements of a PULNIX model TM-34K camera (384 × 491 pixels). The plane was assumed to be two-dimensional and generated zero width lines.

The simulation was run with the corner feature at different orientation angles β, Figure 3.14. The total length of the simulated intersection lines was 250 to 300 pixels. At a particular value of β, the location of the corner was slightly perturbed then measured. A number of simulated measurements of the corner position near a particular value of β were calculated and statistics of the measurements were generated. The difference between the actual corner orientation and the measured corner orientation is shown in Figure 3.15a. The cartesian displacement between the actual position and measurements are shown in Figure 3.15b. The standard deviations in orientation are below about .15 degrees (3 milliradians) and a standard deviations in displacement less than .004 inches (.1 mm). This preliminary study showed that the accuracies expected from the sensor were on the same order as the specifications in Section 3.4.

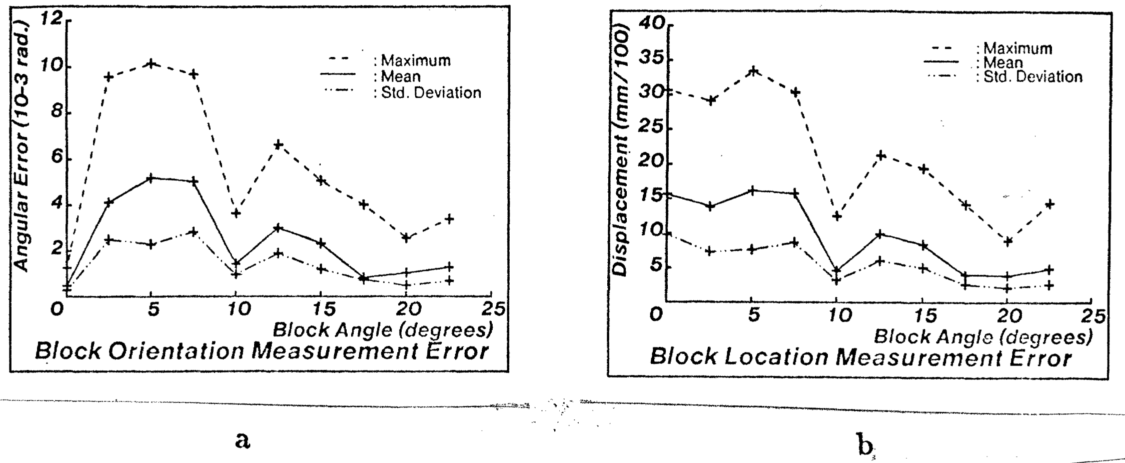


Figure 3.15: Results of the simulated corner localization tests. a. Rotational Accuracy. b. Translational Accuracy.

Feature Localization Using a Light Stripe Vision System

Chapter 4

In this chapter a high-accuracy light-stripe vision system is presented and some mechanisms which generate errors in measuring part positions are analyzed. Errors may be generated from the discrete nature of sensors, inaccuracies in part models, errors in the calibration procedures, inaccurate sensor system model, and system parameter variations due to changing environmental conditions. Quantization errors are analyzed and some techniques for improving measurement accuracy are developed as a result of the analysis.

4.1 Literature Review of Feature Extraction Techniques

Accuracy in Feature Detection

Little attention has been given in the literature to extracting accurate information from visual images. Most vision research deals with qualitative scene analysis (trying to get machine vision systems to do what human vision systems are capable of doing) rather than making accurate measurements. Typical methods used to identify features in an image are Hough transforms and edge detection techniques. Hough transforms were originally developed as a computationally efficient method for detecting lines in an image [17,54,94]. Their use has been extended to nonlinear features as well [18,111,127]. The accuracy of the Hough transform depends upon the resolution of the tessellation of the parameter space; thus, relatively large storage requirements are needed for high precision measurements. For images containing only a few well defined lines, such as a light stripe illuminated scene, Hough transforms are usually not necessary for identifying lines. A large volume of literature which sometimes concerns itself with accuracy issues is the edge and feature detection literature [40,41,49,92,118,121]. MacVicar and Binford [118] claim subpixel edge detection accuracy for a modified Binford-Horn detector although no data is presented. Canny [40,41] derives an edge detector operator which is optimal with respect to three performance criterion:

Good detection. There should be a low probability of both failing to mark real edge points and a low probability of falsely marking non-edge points.

Good localization. Points marked as edge points should be close to the actual edge.

Only one response per edge.

Canny defines a localization metric for a feature detector which is used in Section 4.4 for investigating the accuracy in locating the center of a light stripe.

The accuracy of dimensional measurements from visual images has been studied by groups at General Electric [128,155,156] and SRI International [88]. Mundy and Porter at G.E. determine the accuracy in measuring surfaces with reflectance variations. The technique has been applied to turbine blade inspection. Hill at SRI determines the accuracy of locating binary "blobs" in images based on the number of pixels illuminated by the blob. The accuracy of area calculations are also considered. A probabilistic approach was taken and results were verified with Monte Carlo simulations and laboratory experiments. A similar approach is used to determine the standard deviation in fitted line parameters in Section 4.3.

Using Multiple Measurements

The error in estimating variables from noisy measurements may be decreased by using multiple independent or partially dependent measurements. Additional measurements may come from the same sensor or a completely different source.

Bajcsy and Allen [8,9,15] integrate vision and touch to make measurements of points on the surfaces of objects. First, the outline of the object is determined by a vision system. This information is then used to drive a manipulator fitted with a touch sensor. A model of the object is constructed from the tactile data. Visual information is never directly integrated with tactile information so conflicting data from disparate sources is not dealt with.

Accumulation and propagation of errors in mechanical assemblies was studied by Taylor [189] and Brooks [34]. Taylor propagates geometric errors through a physical model of an assembly. Brooks addresses a similar problem, but uses a symbolic rather than numerical representation. By assuming maximum bounds on the errors, Brooks is able to propagate certain geometric constraints to determine final errors from a number of sources. Both of these geometric error propagation systems assume a maximum error at each source (non-probabilistic) and will give gross over estimates of errors if a significant number of sources are involved.

Optimal estimation theory [64] may be applied to the best fit orientation and displacement estimation problem. The maximum likelihood estimator gives an estimate from overconstraining data weighted by the covariance between the components of the measurement. No prior knowledge about the position of the object being measured is assumed. A Kalman filter technique [64] may be used to optimally update a current estimate from subsequent independent measurements. Durrant-Whyte [55] combines information from independent observations to get a minimum-risk best estimate of the state of the environment. A Bayesian approach is used to combine errors and a non-recursive estimate is presented. Differential transformations, as developed in [147], are used to represent small errors in orientation and translation. The possibility of spurious measurements is taken into account and when it is likely that such a measurement occurred, it is rejected. A significant number of measurements must be taken in order to do this reliably. New estimates are propagated through a world model to maintain consistency of the model.

Shekhar, Khatib, and Shimojo [171] use a non-probabilistic method to combine a number of rotation and translation measurements into a single estimate. They use a quaternion representation for rotations and assume a diagonal weighting matrix for the set of measurements; thus, dependence between components of a measurement are ignored. Their results are similar to the maximum likelihood results from optimal estimation theory with diagonal covariance matrices. Smith and Cheeseman [179,180] develop two ways of combining what they call "fuzzy transformations." Compounding two fuzzy transformations increases the uncertainty and merging them decreases the uncertainty. Compounding uses the Jacobian of the resultant transformation (derivatives are with respect to the components of the uncompounded transformations) and the covariance matrices of the uncompounded transformations. Merging uses an extended Kalman filtering result. An example covariance matrix calculation for measurements of the planar position of a mobile robot is given.

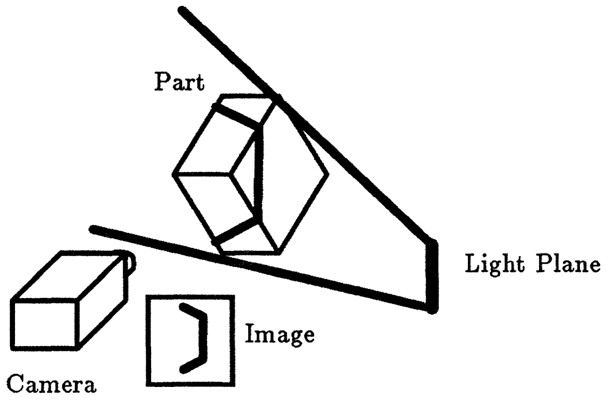


Figure 4.1: Three line segments generated by the intersection of a plane of light and the surfaces of a polyhedral feature may be sensed by a video camera and used to locate a part.

4.2 Light Stripe Part Position Sensor Fundamentals

The part position sensor developed in this thesis uses the light stripe technique to locate parts in six degrees of freedom with respect to a global reference frame. A plane of light is projected across one or more features (such as a corner of a polyhedron or an end of a truncated circular cylinder) of a part (see Figure 4.1). Data from an image taken by a video camera positioned at some disparate angle with respect to the light plane is processed to locate the feature.

The ranging system consists of a line illumination source and a two-dimensional light sensing element whose optical axis is positioned at some finite disparity angle from the plane of the source, Figure 4.2. Triangulation is used to obtain three-dimensional data from the two-dimensional sensing element data. A common light source for light stripe systems is a laser beam which has been passed through a cylindrical lens. The lens diverges the beam in a direction perpendicular to the lens' cylindrical axis. The light sensitive element is often a Vidicon, CCD (charged coupled device) or CID camera (Reference [17] gives a description of each of these cameras).

The light plane is the sole functional illumination source in the scene as viewed by the camera. Light from the line source reflects off surfaces in the scene and

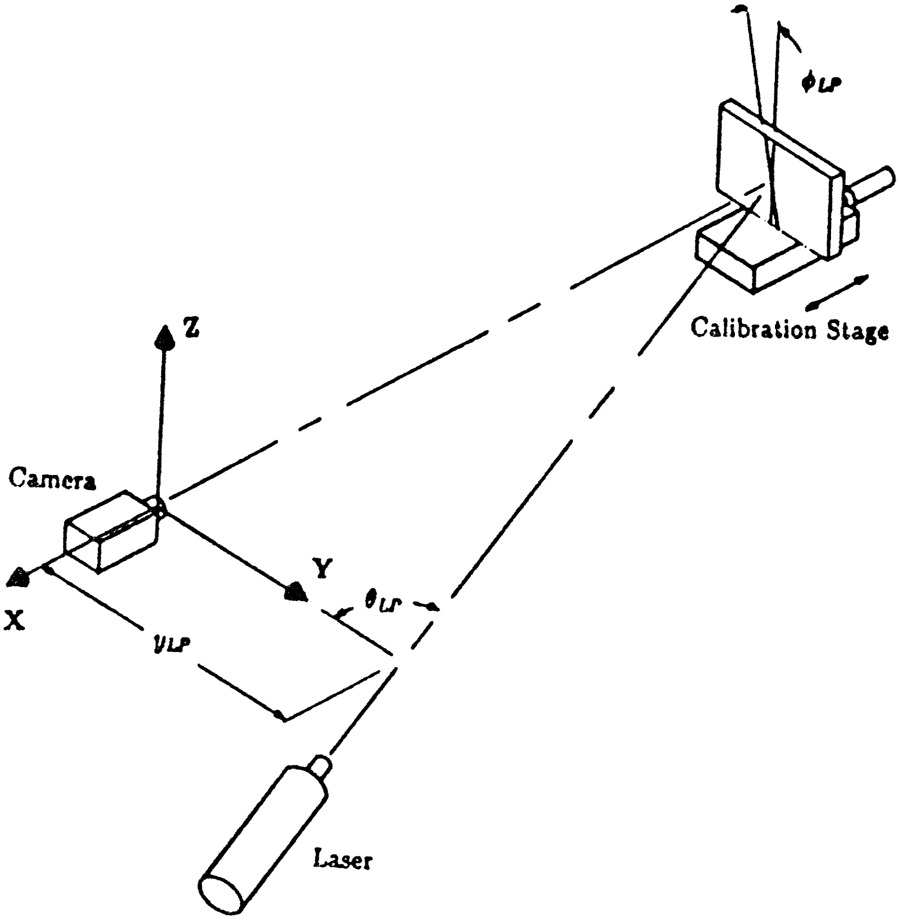


Figure 4.2: Light stripe sensor configuration.

appears on the two-dimensional image plane of the camera. An isomorphism exists between all points in the image and points lying in the light plane in three-space. In order to determine the transformation from the two-dimensional image coordinates to accurate three-dimensional space coordinates, the system must be precisely calibrated (see Section 6.3). During the calibration, the values of three parameters must be determined which locate the light plane with respect to a coordinate frame defined by the camera image plane. A disparity angle, θ_{LP}, a tilt angle, ϕ_{LP}, and an offset, y_{LP}, are the parameters used for defining the light plane location (these are not unique), Figure 4.2.

Light rays from illuminated points on the part's surfaces undergo a perspective projection into the camera; thus, illuminated lines in space remain lines in the image plane, but most other shapes are distorted. Most of our discussion will be limited to objects whose surfaces are planar (actually only those surfaces being sensed need be planar); thus, the intersection curves between the light plane and part surfaces are lines.

4.2.1 Review of Elementary Optics

Only optical relationships needed for the subsequent analysis are included in this section. The reader is referred to an introductory optics text such as [17,93,103,178] for more detailed explanations.

We model the camera lens using a thick lens model [17,93,103,178]. The image plane is positioned behind the rear principle plane and the object before the front principle plane. Light rays which go through the front principle plane at the optical axis, pass through the rear principle plane at the optical axis, then onto the image plane. For a lens with no distortion, the angle of a light ray with respect to the optical axis is of the same magnitude as the angle of the ray as it leaves the rear principle plane. Spherical or other abberations [178] may change the direction of the ray as it leaves the rear nodal point. From Figure 4.3 we may determine the relationship between the object size, z_O, and the image sizes, z_I. For a lens with no distortions, $\alpha_1 = \alpha_2$

$$\tan \alpha_1 = \tan \alpha_2 = \frac{z_O}{-x} = \frac{z_I}{f_0}$$

so

$$z_I = -\frac{z_O f_0}{x}. \tag{4.1}$$

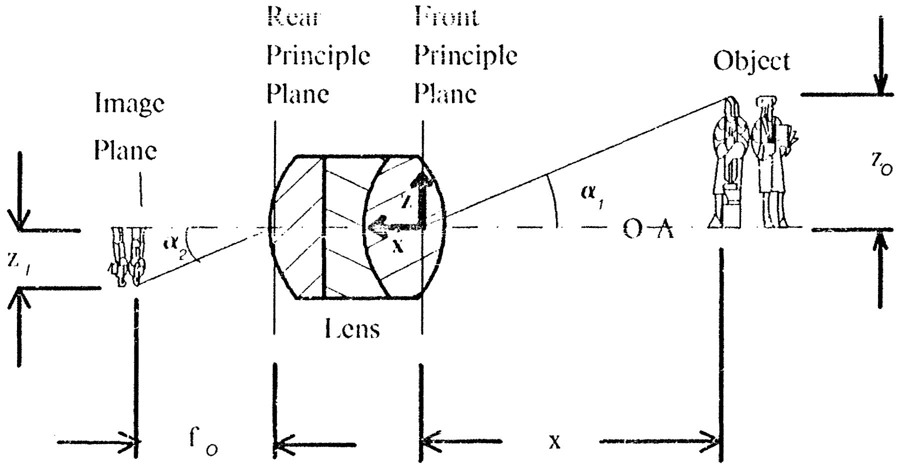


Figure 4.3: Parameters for modeling a thick lens.

The minus signs are due to the orientation of the x axis.

4.2.2 Determining World Coordinates from Sensor Data

Here we derive the coordinates of a point in space as a function of the image plane coordinates and the light plane and camera calibration parameters. Some authors have used a matrix description for the geometry of a light stripe system [3,136]. Separate equations are maintained here. An equation similar to Equation (4.1) describes the size of the image in the y direction

$$y_I = -\frac{y_O f_0}{x} \tag{4.2}$$

We obtain the x coordinate from the geometric relationships shown in Figure 4.2

$$x = x_0 + \tan\theta_{LP}\left(y - z\frac{\tan\phi_{LP}}{\sin\theta_{LP}}\right) \tag{4.3}$$

where x_0 is the position along the optical axis where the light plane crosses,

$$x_0 = -y_{LP}\tan\theta_{LP}.$$

In general $x_0 < 0$. Combining Equations (4.1) (4.2), and (4.3), we obtain

$$x = \frac{x_0}{1 + \frac{\tan\phi_{LP}}{\cos\theta_{LP}}\frac{z_I}{f_0} + \tan\theta_{LP}\frac{y_I}{f_0}} \tag{4.4}$$

$$y = -\frac{\frac{y_I}{f_0} x_0}{1 + \frac{\tan\phi_{LP}}{\cos\theta_{LP}} \frac{z_I}{f_0} + \tan\theta_{LP} \frac{y_I}{f_0}} \tag{4.5}$$

$$z = -\frac{\frac{z_I}{f_0} x_0}{1 + \frac{\tan\phi_{LP}}{\cos\theta_{LP}} \frac{z_I}{f_0} + \tan\theta_{LP} \frac{y_I}{f_0}}. \tag{4.6}$$

These equations describe the position of a point in space (x, y, z) as a function of the coordinates of the corresponding point in the image plane (y_I, z_I).

Calculation of Three-Dimensional Vectors from the Light Stripe Image

The measurements obtained from the image of the intersection of a light plane and a polyhedral feature is a set of line parameters, $[m_{I,i}, b_{I,i}]$, which are defined by the equation of the line (refer to Figure 4.4)

$$y_I = m_{I,i} z_I + b_{I,i}. \tag{4.7}$$

Using Equations (4.4), (4.5), (4.6), and (4.7) we may obtain expressions for the vectors along the light stripes in real space, $\vec{l}_i$, and vectors from the origin to the intersection of the light stripes and the $z = 0$ plane, $\vec{b}_i$.

$$\vec{l}_i = \left\{ \begin{matrix} m_i \tan\theta_{LP} \\ m_i \\ 1 \end{matrix} \right\} \tag{4.8}$$

and

$$\vec{b}_i = \left\{ \begin{matrix} \frac{x_{LP} \tan\theta_{LP}}{1 + \frac{b_{I,i}}{f_0} \tan\theta_{LP}} \\ \frac{x_{LP} \frac{b_{I,i}}{f_0} \tan\theta_{LP}}{1 + \frac{b_{I,i}}{f_0} \tan\theta_{LP}} \\ 0 \end{matrix} \right\} \tag{4.9}$$

where

$$m_i = \frac{m_{I,i} - \frac{b_{I,i}}{f_0} \frac{\tan\phi_{LP}}{\cos\theta_{LP}}}{1 + \frac{b_{I,i}}{f_0} \tan\theta_{LP}}. \tag{4.10}$$

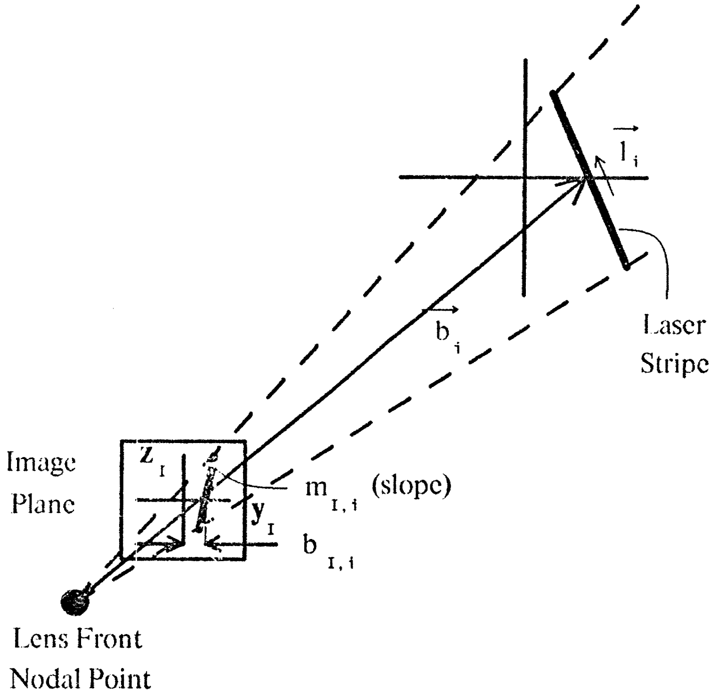


Figure 4.4: Parameters for a line in the image plane and vectors specifying the light stripe.

4.3 Locating Straight Line Features in Quantized Images

4.3.1 Introduction

The precision of a least squares technique in locating the orientation and position of thin line segments in the presence of quantization noise is investigated. A similar technique to Hill's [88] is used to determine the accuracy in finding these parameters. Accuracy in locating two dimensional images, edges, thin lines, and curve segments are important for precise manipulation with endpoint sensing [190], robot calibration [153], integration of vision with precise mechanical and electronic assembly tasks (see examples in Chapter 5 and [22,172]), surface inspection [130,156], and vision feedback servo control [199].

We are interested in quantifying the apparent positional shift in measurements of the location of objects due to the discrete nature of data from many optical sensing systems. Straight line segments are of primary interest because they are surface bounding curves for polyhedra and are frequently generated from data from light stripe ranging sensors. Aside from sparse range data techniques [63,76,77,78,79], systems in the literature capable of determining part pose from light stripe data construct a depth map of the entire object [28,60,91,174]. If geometric parameters of the light plane are used in conjunction with an accurate part model, only a single scan of a non-occluded polyhedron is necessary to determining its location and orientation. This is possible if the *a priori* orientation of the object is known within a certain range. If the light plane intersects the polyhedron across three independent surfaces, the object may be accurately located in six degrees of freedom (see Section 4.7). The accuracy with which a polyhedron may be located in three dimensions can be derived from the results for the two degree of freedom line.

4.3.2 Errors in Fitting Linear Parameters to Discretized Data

Data generated by straight-line features appear as discrete points of various intensities located within some width of a central axis in the image plane of a discrete array sensor. These points can be processed to find the best fit line through them using a least squares technique [2,20]. This section explores the accuracy with which straight line features may be found. The variables used in the subsequent analysis are listed in Table 4.1.

Variable	Description
$\mathcal{L}$	Line target
$\mathcal{L}_{image}$	Image of $\mathcal{L}$ in the camera's image plane
N	Number of illuminated pixels.
$(\bar{x}_i, \bar{y}_i)$	Coordinates of the center of the ith illuminated pixel
(x_i, y_i)	Coordinates (probabilistic) of points lying on $\mathcal{L}_{image}$
δy_i	Probabilistic distance in the y direction from the center of the i^{th} illuminated pixel to $\mathcal{L}_{image}$
θ	Orientation angle of $\mathcal{L}$ measured counterclockwise from the x axis
y_0	y axis intercept of $\mathcal{L}$
$\widehat{\theta}$, $\widehat{y_0}$	Least squares estimators for θ and y_0
$\bar{\widehat{\theta}}$, $\bar{\widehat{y_0}}$	Estimators for θ and y_0 from discrete data $(\bar{x}_i, \bar{y}_i)$
$\Delta\bar{\widehat{\theta}}$, $\Delta\bar{\widehat{y_0}}$	Difference between estimated line parameters $(\bar{\widehat{\theta}}, \bar{\widehat{y_0}})$ and actual parameters (θ, y_0)
L_{pixel}	Length of each square pixel
σ_y^2	Variance of δy_i for all $i = 1, \ldots N$
x_{cm}	Distance from the center of $\mathcal{L}_{image}$ to the y axis measured parallel to the x axis

Table 4.1: Nomenclature for line parameter error analysis.

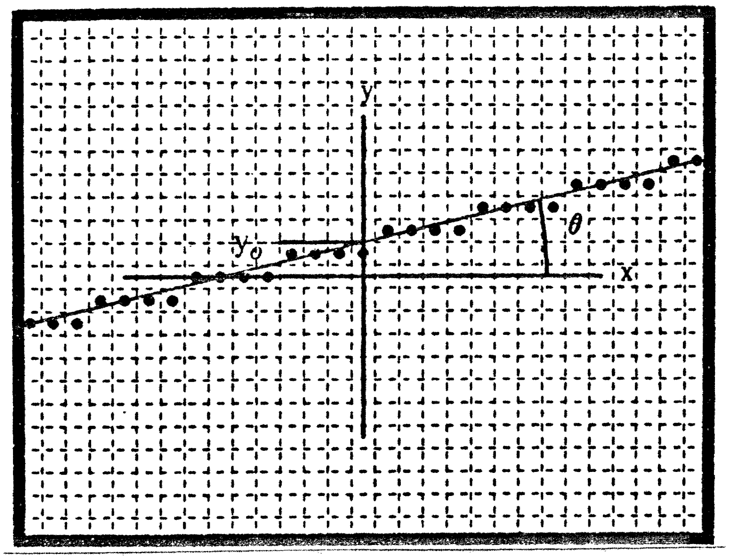


Figure 4.5: Parameters for the image of a line (infinitely thin).

Theoretical Errors from Linear Least Squares Approximations

For the following analysis it is assumed that an image is generated from a single line (infinitely thin), $\mathcal{L}$, which lies in a plane parallel to the image plane. An approximation to an infinitely thin line might be a bounding edge of a surface or the intersection curve of a thin plane of light and a planar surface. The line is located at orientation θ from the horizontal (x) axis and passes over the y axis at intercept y_0. For simplification, the projection of $\mathcal{L}$ into the image plane produces an image, $\mathcal{L}_{image}$ identical to $\mathcal{L}$ (that is, the angle and intercept remain θ and y_0 respectively – see Figure 4.5). A typical output from a CCD (charge coupled device) video camera (the only type being considered here) is an analog signal constructed from a number of intensity readings from discrete pixels in a rectangular grid. For this analysis, we assume that the image has been thresholded and transformed into a binary array of square elements with length L_{pixel}.

For geometric reasons, when $\theta \in [\frac{\pi}{4}, \frac{3}{4}\pi]$, one pixel per row is illuminated and when $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, only one pixel for every column is illuminated. We define an illuminated pixel as the one in a particular row (or column) whose border circumscribes the longest segment of $\mathcal{L}_{image}$. Although this definition loses some positional information from the array, it simplifies the subsequent analysis. Later in this section corrections are suggested which preserve more of the available positional information.

The sensor data used for curve fitting is a list of pixel center coordinates for the illuminated elements $(\bar{x}_i, \bar{y}_i)$ $i = 1, \ldots N$. A "best fit" orientation and intercept ($\hat{\theta}$ and $\widehat{y_0}$ respectively) may be found in the least squares sense by minimizing the sum of the squares of the distances from the best fit line to data points $(\bar{x}_i,\ \bar{y}_i)$ with distance measured one of three ways:

A: *Minimize distances parallel to the y axis.*

B: *Minimize distances parallel to the x axis.*

C: *Minimize distances perpendicular to the line.*

Each of these gives a slightly different result for $\hat{\theta}$ and $\widehat{y_0}$ and the best choice is not immediately apparent.

Selection of the most appropriate metric to minimize depends upon the gross orientation of $\mathcal{L}$. As a result of our definition of an illuminated pixel, (one pixel for every y coordinate is illuminated when $\theta \in [\frac{\pi}{4}, \frac{3}{4}\pi]$, and one pixel for every x coordinate is illuminated when $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$), we can choose an x coordinate in the first case which is deterministic ($\bar{x}_i$ will always correspond to a point on the line within that pixel – see Figure 4.6a), and a y coordinate in the second case which is deterministic. With these choices of coordinates, we should minimize errors in probabilistic coordinates y_i in the first case (metric A) and x_i in the second case (metric B).

In the subsequent analysis θ is assumed to lie in the interval $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The least squares estimates for orientation $\hat{\theta}$ and intercept $\widehat{y_0}$ are found by minimizing errors parallel to the y axis [2,20]

$$\hat{\theta} = \arctan \frac{\sum_{i=1}^{N} y_i x_i - \frac{\sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N}}{\sum_{i=1}^{N} x_i^2 - \frac{(\sum_{i=1}^{N} x_i)^2}{N}} \tag{4.11}$$

$$\widehat{y_0} = \frac{\sum_{i=1}^{N} y_i - \tan\hat{\theta} \sum_{i=1}^{N} x_i}{N} \tag{4.12}$$

where (x_i, y_i) are the coordinates of points lying on $\mathcal{L}_{image}$. If (x_i, y_i) were know precisely, the least squares estimates would be identical to the actual line parameters; however, in the interval $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, only precise x_i values are available.

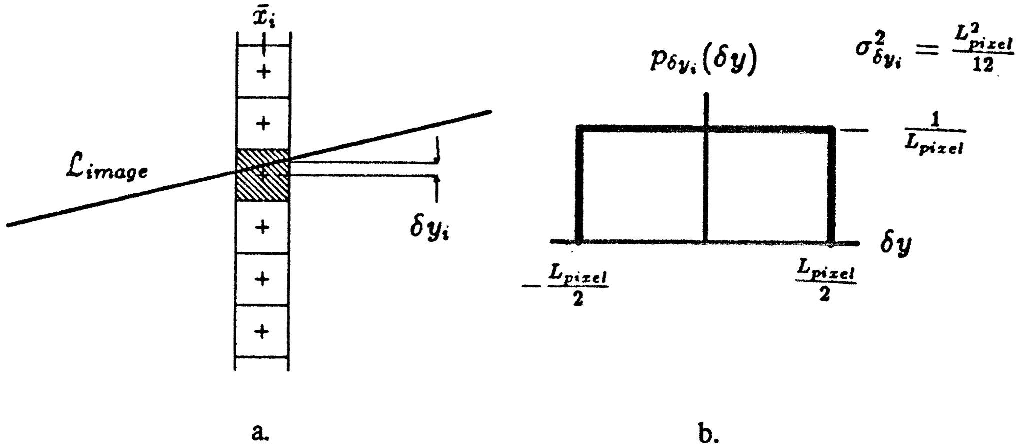


Figure 4.6: Probabilistic location of points on a line. a. Intersection of a column of pixels with a line in the range $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. b. Probability density function for the location of the y coordinate of $\mathcal{L}_{image}$ measured from the center of the illuminated pixel.

We are considering the case where x_i is deterministic and y_i is probabilistic and we desire a metric for the confidence of the estimated line parameters $\hat{\theta}$ and $\widehat{y_0}$ calculated from the measurements $(\bar{x}_i, \bar{y}_i)$. We shall derive the variance of $\hat{\theta}$ and $\widehat{y_0}$ ($\sigma^2_{\hat{\theta}}$ and $\sigma^2_{\widehat{y_0}}$ respectively) as a function of all $(\bar{x}_i, \bar{y}_i)$'s and the variance of y_i ($\sigma^2_{y_i}$).

Each random variable y_i may be written as

$$y_i = \bar{y}_i + \delta y_i \tag{4.13}$$

where $\bar{y}_i$ is deterministic and δy_i is a random variable with zero mean. At first it might appear that the δy_i's are highly correlated since they all lie on the same line; however, some authors [88,144] have suggested that independence between δy_i's is a good assumption for certain cases. In the case of the straight line, the degree of correlation depends upon the orientation of the line. At some orientations, the values of δy_i may change in an unrelated fashion (slightly correlated) while at other orientations, the values of δy_i may exhibit a periodic pattern (highly correlated). We make the initial assumption that the δy_i's are independent and a supposition that this might not be valid for lines at certain orientations.

If the maximum absolute value of the δy_i's are small, Equations (4.11) and (4.12) can be linearized about the points $(\bar{x}_i, \bar{y}_i)$ by taking the Taylor series expansion, and keeping the first two terms

$$\widehat{\theta} \approx \bar{\widehat{\theta}} + \sum_{i=1}^{N} \left(\frac{\partial \widehat{\theta}}{\partial y_i} \right)_{(\bar{x}_i, \bar{y}_i)} \delta y_i \tag{4.14}$$

$$\widehat{y_0} \approx \bar{\widehat{y_0}} + \sum_{i=1}^{N} \left(\frac{\partial \widehat{y_0}}{\partial y_i} \right)_{(\bar{x}_i, \bar{y}_i)} \delta y_i \tag{4.15}$$

where $\left(\frac{\partial}{\partial y_i}\right)_{(\bar{x}_i,\bar{y}_i)}$ is the partial derivative with respect to y_i evaluated at $(\bar{x}_i, \bar{y}_i)$ and

$$\bar{\widehat{\theta}} = \arctan \frac{\sum_{i=1}^{N} \bar{y}_i \bar{x}_i - \frac{\sum_{i=1}^{N} \bar{x}_i \sum_{i=1}^{N} \bar{y}_i}{N}}{\sum_{i=1}^{N} \bar{x}_i{}^2 - \frac{(\sum_{i=1}^{N} \bar{x}_i)^2}{N}} \tag{4.16}$$

$$\bar{\widehat{y_0}} = \frac{\sum_{i=1}^{N} \bar{y}_i - \tan \bar{\widehat{\theta}} \sum_{i=1}^{N} \bar{x}_i}{N}. \tag{4.17}$$

For all δy_i independent of one another [53]

$$\sigma^2_{\widehat{\theta}} \approx \sum_{i=1}^{N} \left(\frac{\partial \widehat{\theta}}{\partial y_i} \right)^2_{(\bar{x}_i, \bar{y}_i)} \sigma^2_{\delta y_i} \tag{4.18}$$

$$\sigma^2_{\widehat{y_0}} \approx \sum_{i=1}^{N} \left(\frac{\partial \widehat{y_0}}{\partial y_i} \right)^2_{(\bar{x}_i, \bar{y}_i)} \sigma^2_{\delta y_i}. \tag{4.19}$$

Substituting Equations (4.11) and (4.12) into Equations (4.18) and (4.19) respectively then making the assumption that all δy_i have the same variance σ_y^2, and setting x_i to $\bar{x}_i$, we obtain

$$\sigma^2_{\widehat{\theta}} \approx \frac{1}{(1 + \bar{m}^2)^2} \frac{\sigma_y^2}{\left(\sum_{i=1}^{N} \bar{x}_i{}^2 - \frac{(\sum_{i=1}^{N} \bar{x}_i)^2}{N} \right)} \tag{4.20}$$

$$\sigma^2_{\widehat{y_0}} \approx \frac{1}{N} \left(1 + \frac{\frac{(\sum_{i=1}^{N} \bar{x}_i)^2}{N}}{\sum_{i=1}^{N} \bar{x}_i{}^2 - \frac{(\sum_{i=1}^{N} \bar{x}_i)^2}{N}} \right) \sigma_y^2 \tag{4.21}$$

where

$$\bar{m} = \frac{\sum_{i=1}^{N} \bar{y}_i \bar{x}_i - \frac{\sum_{i=1}^{N} \bar{x}_i \sum_{i=1}^{N} \bar{y}_i}{N}}{\sum_{i=1}^{N} \bar{x}_i{}^2 - \frac{(\sum_{i=1}^{N} \bar{x}_i)^2}{N}}.$$

Equations (4.20) and (4.21) give the variances of the fitted line parameters as a function of the data points $(\bar{x}_i,\ \bar{y}_i)$ and the variance of the location of the y

coordinate of the line in any column containing an illuminated pixel (similar results are obtained in [20]).

If columns $\bar{y}_i$ have an illuminated pixel for all $i = 1, \ldots N$ then the expressions for the variance of $\widehat{\theta}$ and $\widehat{y_0}$ become

$$\sigma^2_{\widehat{\theta}} \approx \frac{1}{(1+\bar{m}^2)^2} \frac{12\sigma_y^2}{N^3 - N} \tag{4.22}$$

$$\sigma^2_{\widehat{y_0}} \approx \left(\frac{1}{N} + \frac{12x_{cm}^2}{N^3 - N} \right) \sigma_y^2 \tag{4.23}$$

where $x_{cm} = \frac{\sum_{i=1}^N \bar{x}_i}{N}$ is the center of the line segment in the x direction.

These results describe the accuracy with which a line may be found from a least squares technique as a function of the approximate slope ($\bar{m}$), the orthographic projection of the length of the line (in pixels) onto the x axis (N), the center of the line in the x direction (x_{cm}), and the variance of the location of the y coordinate for any one column of the image (σ_y^2). $\sigma^2_{\widehat{\theta}}$ does not depend on the position of the line segment in the pixel grid, while $\sigma^2_{\widehat{y_0}}$ depends on the distance of the center of the line segment from the y axis. For a line (zero width), the probability density function for mutually independent δy_i is a uniform distribution one pixel in width with a $\sigma^2_{\delta y_i} = \sigma^2_y = \frac{L^2_{pixel}}{12}$ (see Figure 4.6b). The standard deviations $\sigma_{\widehat{\theta}}$ and $\sigma_{\widehat{y_0}}$ are plotted in Figure 4.7 as a function of the number of points N for a line positioned near $\theta = 0$ and with $x_{cm} = \frac{N}{2}$.

The above analysis is valid for a grey scale images of finite width lines as well as binary images of thin lines. If $\bar{y}_i$ is permitted to take on subpixel values, and grey scale levels are used in an intensity weighted "center-of-mass" calculation, $\bar{y}_i$ becomes a better estimate of the actual value y_i (σ_y^2 would be smaller). The reduction in σ_y^2 is a function of the width of the line and the resolution (in intensity measurements) of the camera.

We now consider a bound for the line parameters assuming that all y_i are perfectly correlated. Because we have very little knowledge as to where a line is located on the pixel array, we assume that the *a priori* joint probability density function, $p_{\theta,y}(\theta_a, y_a)$, for orientation θ and position of the line center y is uniform, Figure 4.8a. We take the *a priori* bounds on the orientation and location to be such that all lines must be within an envelope one pixel wide by N pixels long ($\theta \in [-\frac{1}{N}, \frac{1}{N}]$ and $y \in [-\frac{L_{pixel}}{2}, \frac{L_{pixel}}{2}]$). The marginal density functions $p_\theta(\theta_a)$ and

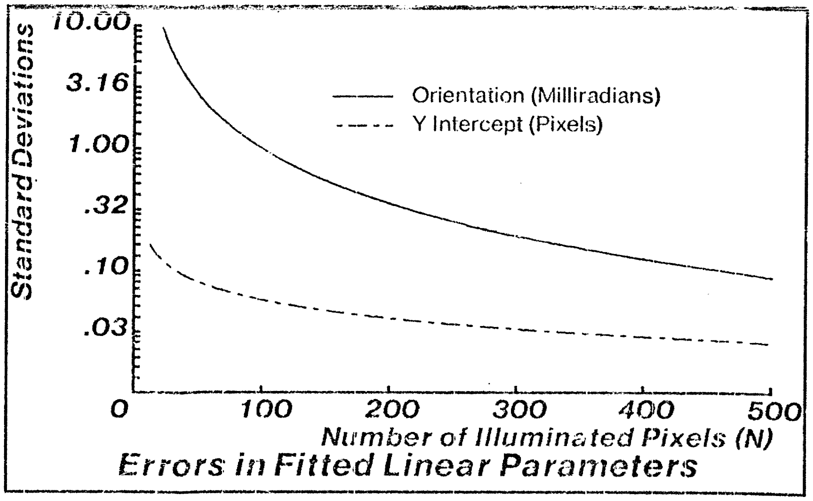


Figure 4.7: Standard deviations for orientation and y intercept estimates as a function of the number of illuminated pixels.

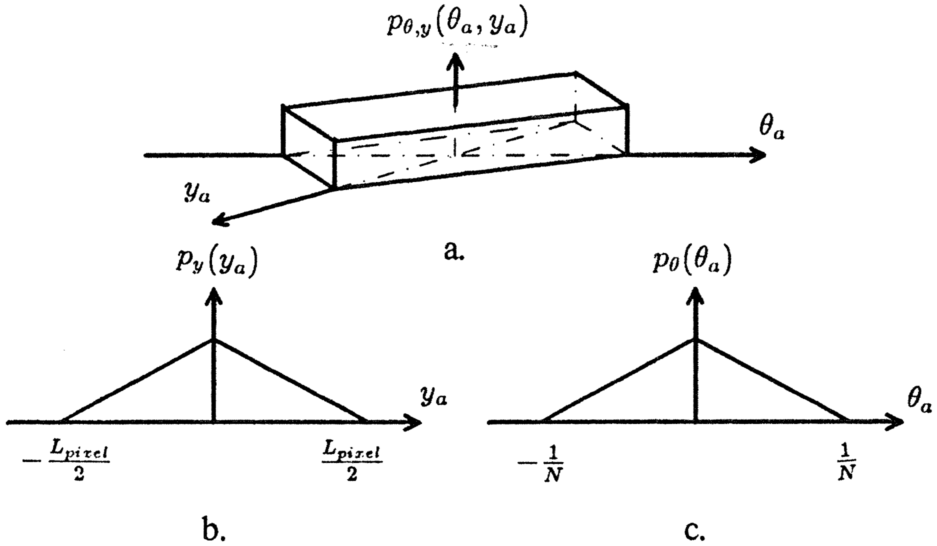


Figure 4.8: Probability distributions of line parameters. a. Joint probability density function for the orientation and y location of a line. b. y location marginal density function. c. Orientation marginal density function.

$p_y(y_a)$ may be calculated from $p_{\theta,y}(\theta_a, y_a)$ (see Figure 4.8b-c)

$$p_\theta(\theta_a) = \int_y p_{\theta,y}(\theta_a, y_a)dy_a, \qquad p_y(y_a) = \int_\theta p_{\theta,y}(\theta_a, y_a)d\theta_a.$$

The variances of the resulting triangular distributions are

$$\sigma_\theta^2 = \frac{1}{6N^2} \tag{4.24}$$

$$\sigma_y^2 = \frac{L_{pixel}^2}{24}. \tag{4.25}$$

Errors for Linear Least Squares Approximations from Simulation and Experimental Tests

A study investigating the accuracy of the linear least squares fitting technique as a function of the orientation and location of a line target was performed by way of computer simulation and an experimental test using a CCD video camera. The target used for the tests was a straight-line step discontinuity in intensity, Figure 4.9. Line $\mathcal{L}$ defines the location and orientation of the edge.

Errors as a Function of Image Orientation from Simulation Tests

The least squares estimator from the discrete data, $\hat{\bar{\theta}}$, is compared to the actual orientation of the line, θ. Measurements of the errors in the estimator, $\Delta\hat{\theta} = \theta - \hat{\bar{\theta}}$, are made at various orientations of the line. For a selected orientation, $\Delta\hat{\theta}$ is locally studied by generating lines at random orientations within a small range. Sample statistics from 30 trials are calculated and used as a basis for comparisons. Figure 4.10 shows the results of the study with statistics generated in each of eighteen equally spaced orientations from zero to forty-five degrees. The sample standard deviations for each set of thirty trials (each trial is at a different random orientation within a 2.5 degree range) with a line 128 pixels long are plotted for each interval along with the theoretical result [Equation (4.22)]. A mirror-image plot is generated above forty-five degrees for fitted parameters calculated using metric B. Singularities in $\Delta\hat{\theta}$ occur at slopes near 0, $\frac{1}{4}$, $\frac{1}{3}$, $\frac{1}{2}$, $\frac{3}{4}$, and 1. In these areas, measurements of the location and orientation of $\mathcal{L}_{image}$ are not as accurate because there is more space for $\mathcal{L}$ to translate and rotate before pixels are caused to

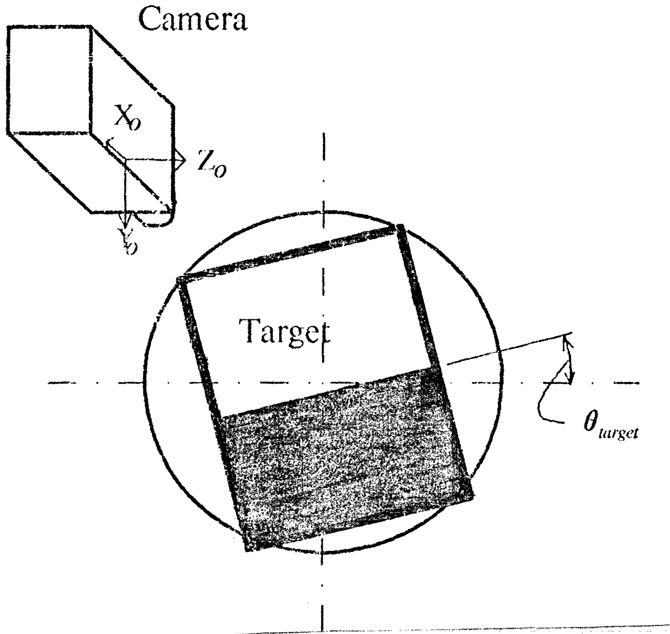


Figure 4.9: Target and camera arrangement for computer simulation and experimental tests.

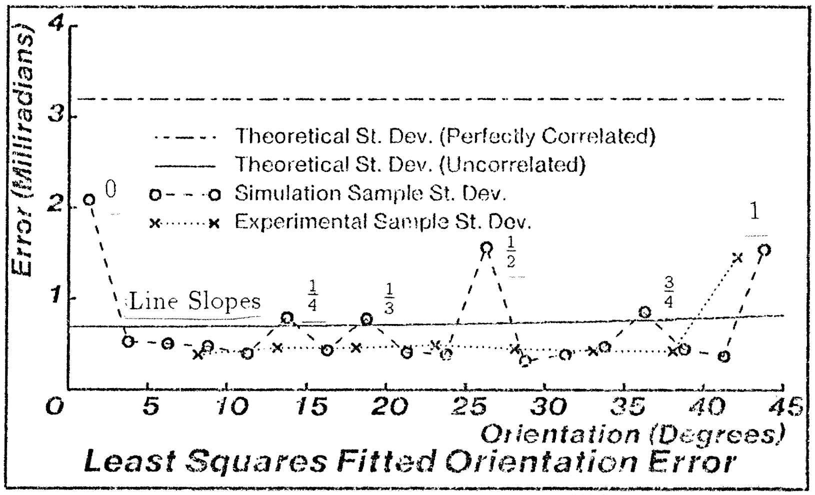


Figure 4.10: Accuracy of the least squares fitting routine for discretized lines as a function of their orientation. Theoretical, computer simulation and experimental results are shown.

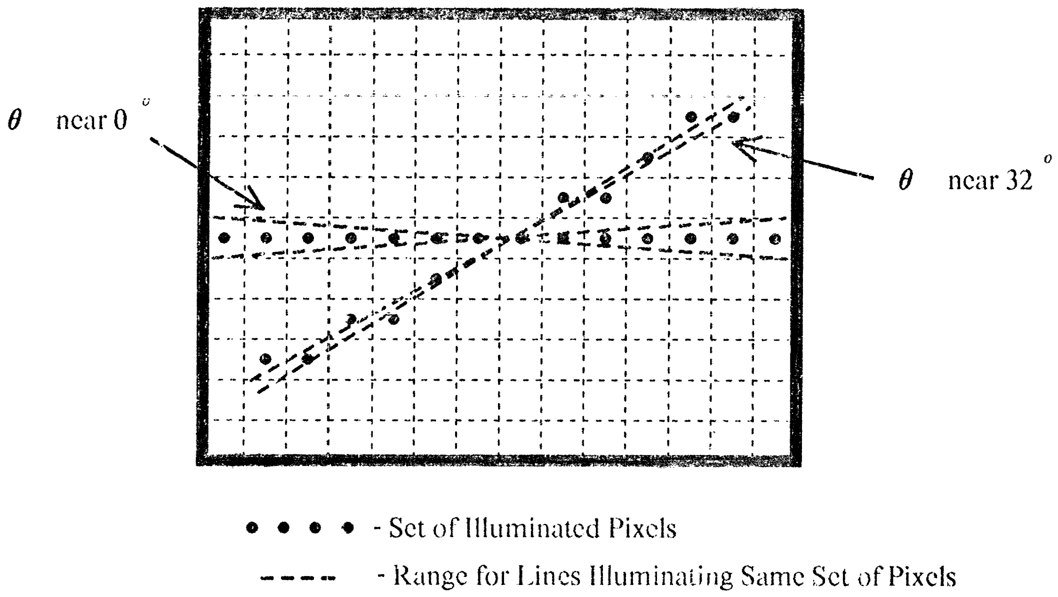

θ near 0°
θ near 32°
● ● ● ● - Set of Illuminated Pixels
- - - - - Range for Lines Illuminating Same Set of Pixels

Figure 4.11: Range of motions of lines before change in state of sensor occurs.

change state. (see Figure 4.11). Statistics were collected over a finite orientation range to identify the areas most prone to errors. The theoretical result is an accurate bound except at the singularities near pixel ratios (slopes) of 0, $\frac{1}{2}$ and 1. The theoretical result does not predict these singularities because the mutual independence assumption is not valid there. The experimental sample standard deviations at all orientations are, however, bounded by the theoretical result for the perfectly correlated case [Equation (4.24)].

Errors as a Function of Image Orientation from Experimental Tests

The theoretical and simulation results were verified by tests performed with a CCD camera (Hitachi model KP-120). The camera was externally synced (horizontal and vertical) to a frame grabber which was interfaced to a Symbolics 3600 Lisp Machine. A machinist's rotary table and the bed of a vertical milling machine were used as accurate positioning stages for the edge target (see Figure 4.12). Tests were performed at various target orientations by analyzing binary arrays generated from thresholded 8 bit frame grabber arrays. At each orientation, thirty trials were performed. The target was translated about $\frac{1}{12}$ of a pixel for each trial. Results are shown in Figure 4.10. The experimental results are close to the computer simulation results except for an apparent absence of singularities. This is due to the difference in test procedures. In the computer simulations, the target was randomly oriented within a 2.5 degree range and had a larger probability of being near a singularity orientation than the target in the experimental test.

Error Sources

Careful camera calibration was important for accurate line parameter fitting. In addition to finding the affine transformation between the camera and target coordinate frames, it was necessary to accurately determine a height to width ratio of the frame-grabbed image. The ratio may differ from 1 if the pixel rate clocks in the camera (not externally synced) and the frame grabber differ. Most video cameras do not have an external pixel clock syncing facility and are subject to error. These errors were found to be temperature sensitive. Table 4.2 gives sources of errors in measuring line parameters found during the test bed tests. The size of the check mark corresponds to the estimated contribution of each error source. All errors were small, but the most significant were due to pixel aspect ratio inaccuracies, changes in camera aperature light source position, and significant variation in camera temperature which was brought about by moving the light source near the camera. Lens abberations also affected the measurements, but these were not investigated.

Errors as a Function of Image Length from Simulation Tests

Sample statistics from a number of trials performed for lines of various lengths randomly oriented within a range 5 to 10 degrees from the horizontal are plotted along with the theoretical bound for the standard deviation in orientation [Equation (4.22)] in Figure 4.13. The sample standard deviation of the angular error as a function of the number of illuminated points in the image is bounded by the theoretical result. At this orientation, the sample standard deviation is an average of 62 percent of the theoretical result.

Error Source	Contribution
Pixel Aspect Ratio	√
Pixel Location	√
Camera Focus	√
Camera Aperature	√
Light Source Position	√
Light Source Intensity	√
Light Source Position	√
Linearity of Camera Receptors	√
Table Positioning Errors	√
Table-Camera Alignment	√
Variation in Camera Temperature	√
Non-Linearities in Lens	?
Reflectance of Target	√
Camera Blossoming	√
Uniform Light Source	√

Table 4.2: Experimental error sources for measuring straight-line features.

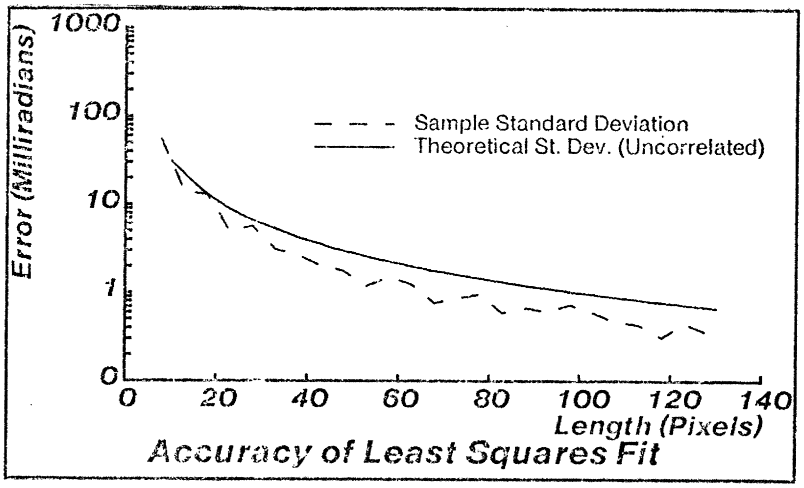


Figure 4.13: Dependence of the least squares algorithm on the length of an image of a line segment at orientations is between 5 and 10 degrees.

4.3.3 Conclusions

For a discretized image of a thin line, least squares estimators which minimize distances parallel to the pixel axis most nearly perpendicular to the line image should be employed. This minimizes errors in the direction of greatest uncertainty since the line always passes through the center of the pixels along the axis most nearly parallel to the line.

Standard deviations in least squares fitted line parameter errors are bounded by curves proportional to $\frac{1}{1+\bar{m}^2}\sqrt{\frac{1}{N^3-N}}$ in orientation and $\sqrt{\frac{1}{N}+\frac{12x_{cm}^2}{N^3-N}}$ in translation where $\bar{m}$ is the approximate slope of the line, N is the number of data points along the line and x_{cm} is the x coordinate of the center of the line. Certain singularity configurations of the line produce relatively large errors in fitted parameters which are not modeled by these expressions. These singularities occur at orientations and locations where the inter-column (row) correlation of the distances being minimized in the least squares algorithm are fairly high. For most orientations, however, these distances may be considered mutually independent.

4.4 Single Row Subpixel Localization of Light Stripe Features

The accuracy in estimating line parameters may be improved by using the additional information contained in a finite width line generated by a stripe source. Better accuracy may be attained by decreasing the errors in locating points used in the least squares line fitting. Three techniques for locating points along a light stripe are explored: thresholding, center of area, and match-filtering/peak-detection.

4.4.1 Intensity Profile of the Light Stripe

In a plane perpendicular to the projector axis, the intensity of the light sheet is a function of the distance from the center of the stripe, Figure 4.14a. The shape of the intensity is the square of the Fourier transform of the source. This is due to diffraction at the source. We assume that the light source is finite in one direction and infinite in the other two orthogonal directions. The Fourier transform of the source is a one-dimensional Fourier transform across the finite width, since the Fourier transform along the infinite direction of the source is a pulse at the origin.

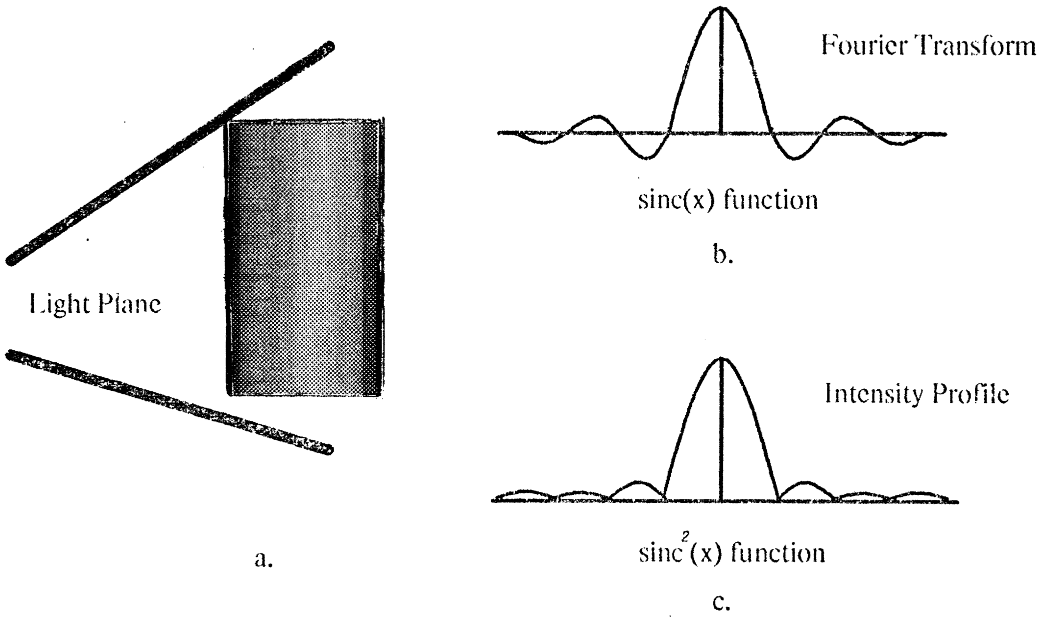


Figure 4.14: Intensity profile of a light stripe.

The one-dimensional Fourier transform of a uniform source is a $sinc(x)$ function [178],[1] Figure 4.14b; thus, the intensity profile is (Figure 4.14c)

$$I(x) = A_{max} sinc^2(ax) = A_{max} \frac{\sin^2(ax)}{(ax)^2}. \qquad (4.26)$$

When a discrete sensing system like a CCD camera observes the $\sin^2(x)/x^2$ profile, each sensor element integrates the intensity over a finite area. If the sensor elements are rectangular and the light stripe is nearly aligned with one of the edges of the rectangles, then the image produced on the sensor is a sampled smoothed $sinc^2(x)$; that is, a $sinc^2(x)$ convolved with a square pulse then multiplied by a pulse train, Figure 4.15. Figure 4.16 shows the intensity profile of a light stripe from one row of an image from a CCD camera.

[1] $sinc(x) = \frac{\sin(x)}{x}$.

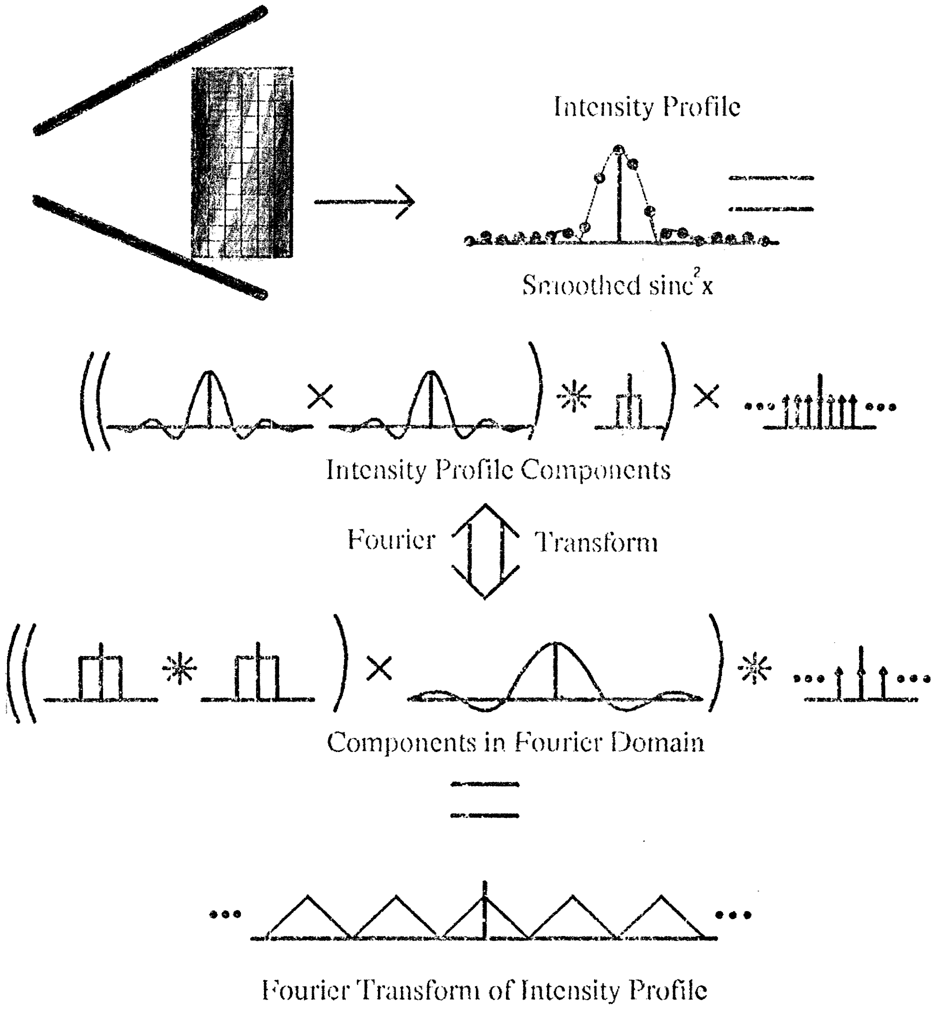

Intensity Profile
Smoothed sinc^{2}x
Intensity Profile Components
Fourier
Transform
Components in Fourier Domain
Fourier Transform of Intensity Profile

Figure 4.15: Fourier transform of a sampled, smoothed $\sin^2(x)/x^2$ function.

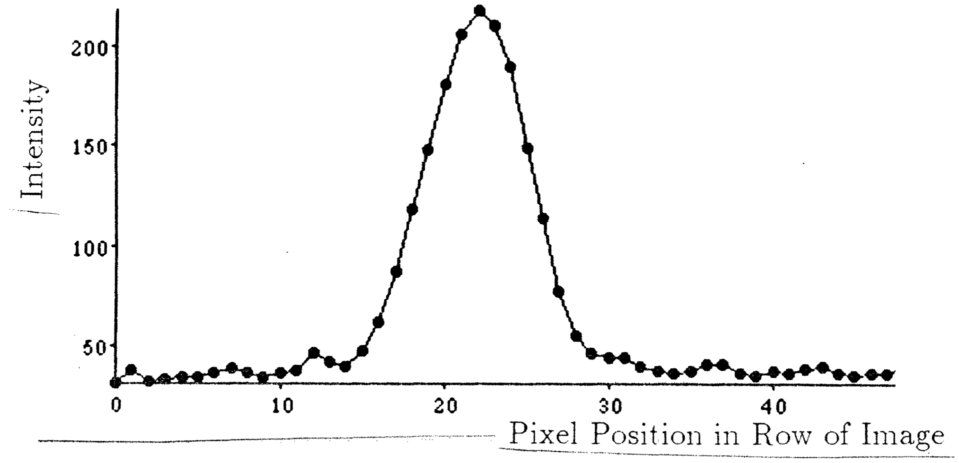


Figure 4.16: Intensity profile of a light stripe as measured by a CCD camera.

Gaussian Approximation of the Intensity Profile

For simplicity in some of the subsequent analyses, we assume that a smoothed $sinc^2(ax)$ function may be accurately modeled as a Gaussian.

$$G(x) = A_0 N(x, \bar{x}, \sigma)$$

where $N(x, \bar{x}, \sigma)$ is the unit normal function

$$N(x, \bar{x}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \tag{4.27}$$

and A_0 is an arbitrary constant. By equating the areas under the curves, $I(x)$ and $G(x)$, and equating the values at $x = 0$, the width of the approximating Gaussian is

$$\sigma_{approx} = \sqrt{\frac{\pi}{2}\frac{1}{a}}$$

and the constant A_0 is

$$A_0 = A_{max}\frac{\pi}{a}$$

A better approximation (empirically determined) of only the center portion of the $sinc^2(x)$ function is a Gaussian of width $0.9\sqrt{\frac{\pi}{2}\frac{1}{a}}$. This approximation is shown in Figure 4.17.

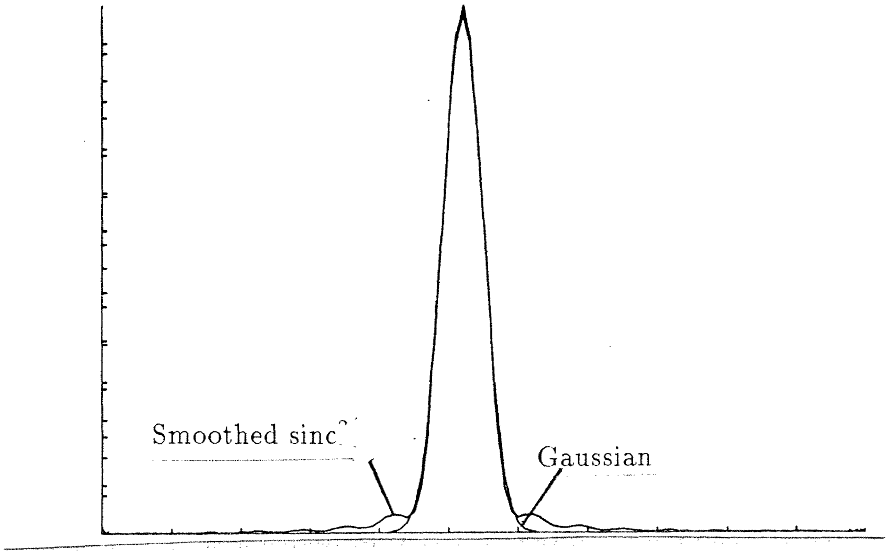


Figure 4.17: Gaussian approximation of a smoothed $sinc^2(x)$ function.

4.4.2 Minimum Sampling Frequency

We determine the Nyquist[2] frequency for the intensity profile by considering its Fourier transform, Figure 4.15. The Fourier transform of a $sinc(ax)$ function is a rectangular pulse of width $\frac{a}{\pi}$. The Fourier transform of the $sinc^2(ax)$ function is the convolution of two rectangular pulses or a triangular pulse of width $2\frac{a}{\pi}$. Multiplying this by the Fourier transform of the thin sampling rectangular pulse (a very wide $sinc(x)$ function) then convolving the result with a pulse train (the transform of a pulse train is another pulse train), we get the desired Fourier transform; a series of nearly triangular pulses of width $2\frac{a}{\pi}$. For no distortion, the corners of the triangles must not touch. The sample spacing of the original waveform must be less than $\frac{\pi}{2a}$ in order to retain all information after sampling.

A similar analysis may be performed for the Gaussian approximation of the intensity profile. To retain all information in the Gaussian, an infinite sampling frequency must be used (the Fourier transform of a Gaussian contains all frequencies); however, since the Fourier transform of a Gaussian (also a Gaussian) approaches 0 amplitude fairly rapidly, we may select a finite sampling frequency which preserves most of the information of the Gaussian. The Fourier transform of a Gaussian of width σ is a Gaussian of width $\frac{1}{2\pi\sigma}$. Figure 4.18 shows the Fourier transform of a sampled Gaussian. Figure 4.19 is a plot of the amount of energy

[2] The Nyquist frequency of a waveform is the minimum frequency at which the waveform may be sampled such that no information is lost due to the sampling process.

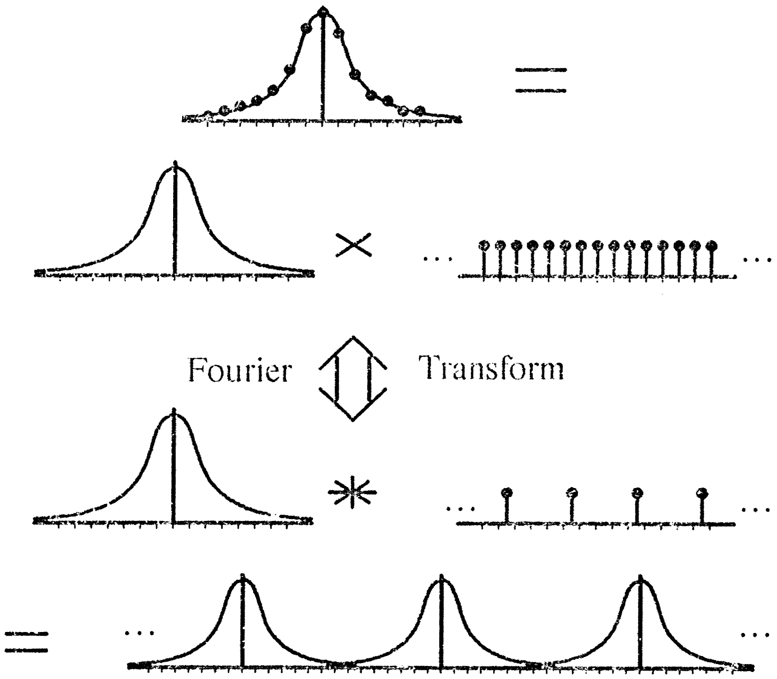


Figure 4.18: Fourier transform of a sampled Gaussian.

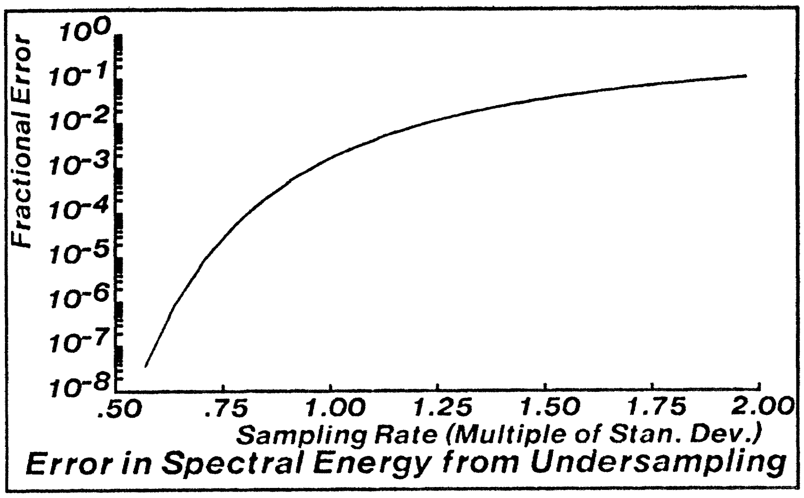


Figure 4.19: Fraction of energy lost in the frequency domain due to undersampling a Gaussian.

lost in the frequency domain due to the overlapping of the Gaussian "tails." For an error of 0.1 percent or less, the Gaussian must be sampled at an intervals closer than one standard deviation.

4.4.3 Adjusting the Width of the Light Stripe

Accurate location of light stripe features are studied to determine what stripe width will produce the smallest errors. In adjusting the width of the light stripe (smoothed $sinc^2(x)$ function), we assume that the power of the light source is also adjustable and in order to obtain the best signal to noise ratio we always wish to raise the power such that the maximum intensity in the image, A_{max}, is a constant and is near the upper intensity limit for the sensor. The equation for a Gaussian with peak value A_{max} is

$$G(x) = A_{max}\sqrt{2\pi}\sigma N(x, \bar{x}, \sigma) \tag{4.28}$$

where $N(x, \bar{x}, \sigma)$ is defined in Equation (4.27).

The following sections discuss the accuracy with which some feature of the intensity profile (such as the peak or the center of area) may be located in the presence of quantization and intensity noise, These results are compared to the results for the case of the line (infinitely thin).

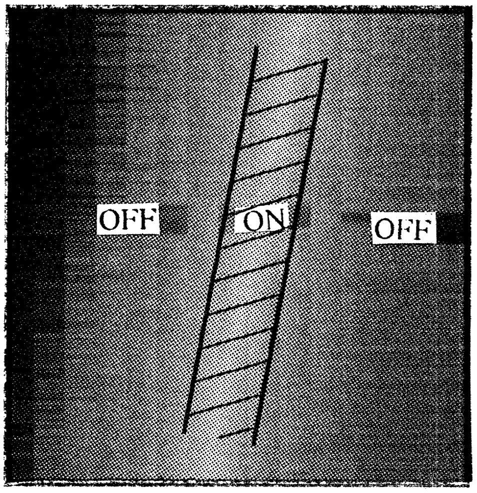


Figure 4.20: Determining the location of an image of a light stripe by thresholding.

4.4.4 Thresholding Technique

One of the simplest and most frequently used image processing techniques is thresholding an image (creating a binary image). Given an intensity profile which extends over a significant number of sensing elements, but whose precise width is unknown, and a reasonable threshold level (say about half of the value of the maximum intensity minus the base noise intensity) then the binary image will have a central "on" region flanked by two "off" regions, Figure 4.20. The only information available in the binary image to locate the light stripe are the two transitions between the off-on-off regions. Since we have assumed that the precise width of the light stripe is unknown, the location of the two edges of the binary image are independent. The location of each of these edges in a particular pixel row may be described by a random variable, x_i. If we assume no knowledge about the shape or location of the original profile, then the probability density function for each of these random variables is a uniform distribution with a width of one pixel. The standard deviation of a one pixel wide uniform distribution is

$$\sigma_x = \sqrt{\frac{L_{pixel}^2}{12}} \tag{4.29}$$

where L_{pixel} is the width of the pixel. If we use the average position of the two edges

$$x_{cm} = \frac{x_1 + x_2}{2}$$

as a metric to locate the light stripe, the variance of this metric (since they are independent) is

$$\sigma_{x_{cm}} = \sqrt{\frac{\sigma_{x_1}^2 + \sigma_{x_2}^2}{4}} = \frac{1}{\sqrt{2}}\sqrt{\frac{L_{pixel}^2}{12}} \qquad (4.30)$$

which is $\frac{1}{\sqrt{2}}$ times the standard deviation in locating the thin line.

4.4.5 Center of Area Technique

In this technique we take advantage of the grey level information in the image. The metric for locating the light stripe is the center of area of the intensity profile

$$x_{cm} = \frac{\sum_i x_i y_i}{\sum_i y_i} \qquad (4.31)$$

where x_i is the location of the i^{th} pixel in a particular row and y_i is the intensity of the i^{th} pixel. We will investigate deviations of this metric from the actual center of a profile due to two sources. The first error source is associated with the noise in the intensity information, and the second source deals with non-symmetric sampling with respect to the peak of the profile.

Errors From Noise in Intensity Levels

If we assume that errors in y_i are small (that is, intensity signal-to-noise is relatively large), we may linearize Equation (4.31).

$$x_{cm} \approx \bar{x}_{cm} + \sum_i \frac{\partial x_{cm}}{\partial y_i} \delta y_i \qquad (4.32)$$

where δy_i is a random variable with zero mean which is the difference between the actual intensity, y_i, and the measured intensity, $\bar{y}_i$ and

$$\bar{x}_{cm} = \frac{\sum_i x_i \bar{y}_i}{\sum_i \bar{y}_i}.$$

For δy_i independent, the variance of the metric is

$$\sigma^2_{x_{cm}} = \sum_i \left(\frac{\partial x_{cm}}{\partial y_i}\right)^2_{\bar{x}_{cm}} \sigma^2_{\delta y_i} \tag{4.33}$$

$$= \frac{\sum_i (x_i - \bar{x}_{cm})^2}{(\sum_i y_i)^2} \sigma^2_{\delta y_i} \tag{4.34}$$

We now assume that the samples are precisely centered about the origin ($\bar{x}_{cm} = 0$) and they are bounded by $\pm\lambda\sigma$; that is,

$$x_i = -\lambda\sigma, -\lambda\sigma + 1, \ldots, 0, \ldots, \lambda\sigma - 1, \lambda\sigma$$

Using this assumption and the fact that the sum of sequential samples along unit spacings of a zero-mean unit Gaussian may be approximated by [53]

$$\sum_{i=y_0}^{y_f} y_i \approx \text{erf}\,(\frac{x_f + \frac{1}{2}}{\sigma}) - \text{erf}\,(\frac{x_0 - \frac{1}{2}}{\sigma}). \tag{4.35}$$

where x_i is the x coordinate which corresponds to y_i, we obtain

$$\sigma^2_{x_{cm}} = \frac{2(\lambda\sigma)^3 + 3(\lambda\sigma)^2 + (\lambda\sigma)}{6\pi\sigma^2[\text{erf}\,(\frac{x_f + \frac{1}{2}}{\sigma}) - \text{erf}\,(\frac{x_0 - \frac{1}{2}}{\sigma})]^2} \frac{\sigma^2_{\delta y_i}}{A^2_{max}}. \tag{4.36}$$

Figure 4.21 shows Equation (4.36) plotted for σ from .5 to 5 samples and for λ from .5 to 4 and noise ratio $(\frac{\sigma^2_{\delta y_i}}{A^2_{max}})$ set to .05.

From Figure 4.21 we see that we want the Gaussian as thin as possible (small σ) and we want to minimize the number of samples (small λ). Note that errors from asymmetric location of the samples relative to the actual peak are considered in the following section.

Errors from Sample Positioning Relative to the Actual Peak

When the center of area metric is used on an intensity profile, the position of the samples relative to the profile may generate errors in the center estimate. In the previous section it was assumed that the samples were centered about the center of mass; whereas, when the location of the profile is unknown *a priori* (this is always the case since the point of making the measurement is to locate the profile), the probability density for the location of the samples is uniformly distributed across

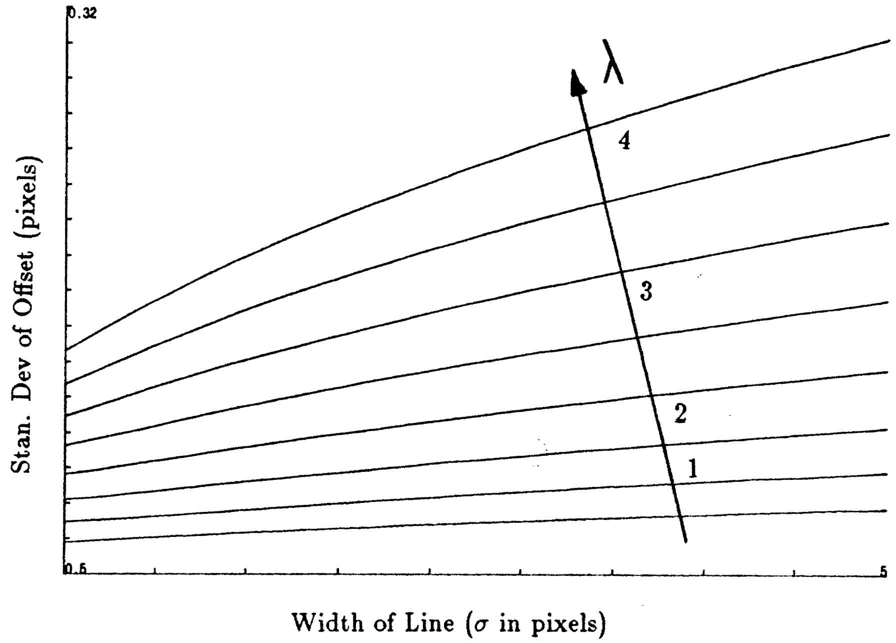


Figure 4.21: Errors in center of area estimate due to noise in intensity levels for a Gaussian intensity profile.

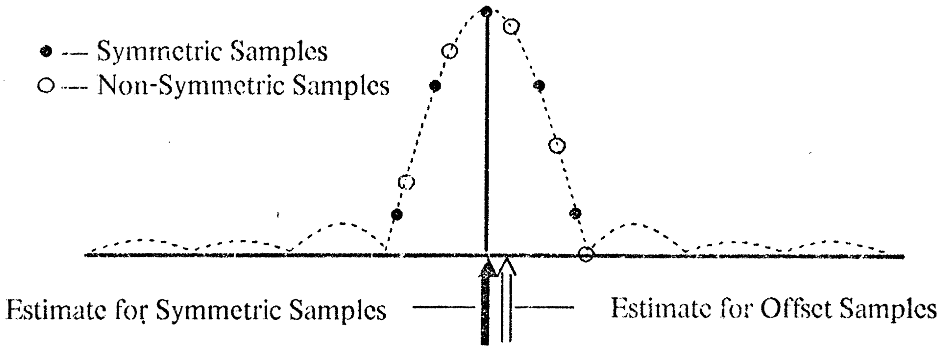

● — Symmetric Samples
○ — Non-Symmetric Samples
Estimate for Symmetric Samples
Estimate for Offset Samples

Figure 4.22: Offsets in sample position from the position of the peak of the intensity profile produces errors in center of area estimates.

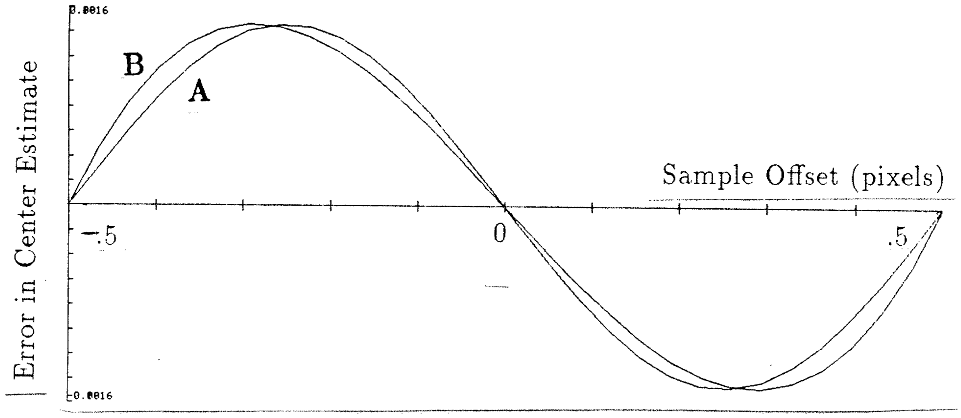


Figure 4.23: Errors in center of area estimate as a function of relative sampling position. Curve A is the numerically calculated error for 10 samples under the main part of the profile and curve B is a sinusoidal approximation to the error.

one pixel width. Figure 4.22 shows the parameters used for the analysis. Assuming a profile of shape $sinc^2x$, the error in center of area estimate [Equation (4.31)] as a function of sampling offset $\epsilon_{samples}$ is shown in Figure 4.23 for the case of 10 samples. Center estimation errors (curve A) from offsets in sample positions from $-\frac{L_{pixel}}{2}$ to $+\frac{L_{pixel}}{2}$ from the peak are shown. Curve B is a sinusoidal approximation to the errors

$$\epsilon_{x_{com}} \approx \epsilon_{max} \sin\left(\frac{4\pi\delta x}{L_{pixel}}\right)$$

where δx is the offset from the peak of the profile to one of the sample positions and ϵ_{max} is the maximum amplitude of the error from the center of area estimate. For computational simplicity, this approximation is used to obtain an expression for the standard deviation of errors. By definition

$$\sigma_{x_{com}} = \sqrt{\int_{-.5L_{pixel}}^{+.5L_{pixel}} \epsilon_{max}^2 \sin^2(\frac{4\pi\delta x_0}{L_{pixel}})\ p_{\delta x}(\delta x_0)\ d\delta x_0} \tag{4.37}$$

where $p_{\delta x}(\delta x_0)$ is the probability density function for the offset δx. This PDF is uniform over the range $\delta x \in [-\frac{L_{pixel}}{2}, +\frac{L_{pixel}}{2}]$ and has a height of $\frac{1}{L_{pixel}}$. Using the relationship

$$\int \sin^2(ax)\ dx = \frac{x}{2} - \frac{\sin(2ax)}{4a},$$

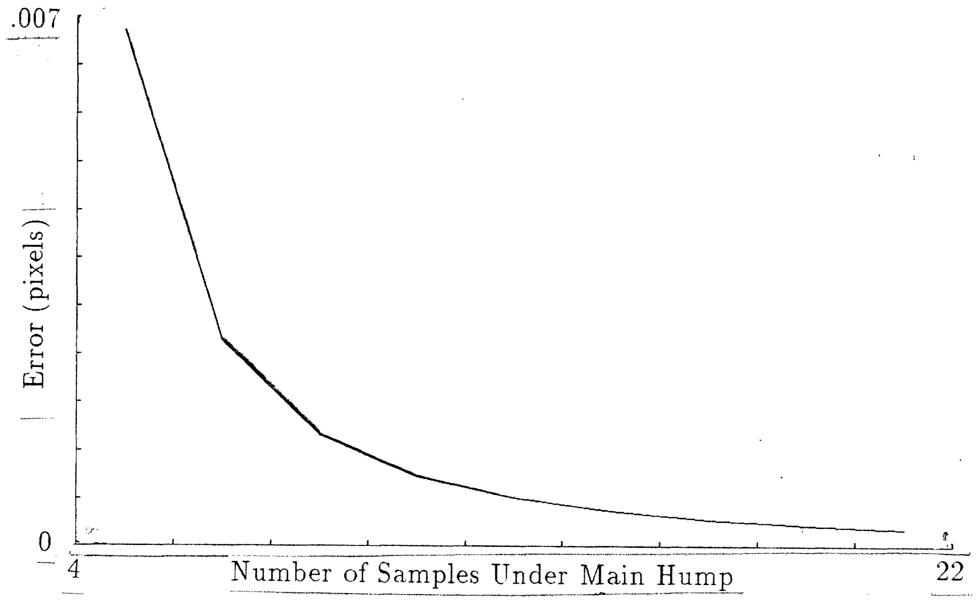


Figure 4.24: Standard deviation of errors due to sample offsets of a $sinc^2(x)$ shaped intensity profile as a function of the number of samples.

Equation (4.37) becomes

$$\sigma_{x_{cm}} = \frac{\epsilon_{max}}{\sqrt{2}} \approx .707\epsilon_{max}.$$

Figure 4.24 shows the standard deviations in errors of center of area estimates for intensity profiles with 5 to 21 samples under the main "hump" (that is $x \in [-\pi, \pi]$ for the $sinc^2(x)$ function).

The relatively small magnitude of the errors from non-symmetric sampling about the peak is due to the truncation of the samples; that is, the samples completely span the main hump but do not go past it. If samples are allowed to extend from the main hump, significant center of area errors may result.

Total Errors From the Center of Area Estimation.

Errors from the three sources (noise in intensity levels, finite sampling width, and sample offsets) are slightly statistically correlated. The coupling is due to the small change in the denominator of Equation (4.36) as the samples are shifted relative to the peak. For simplicity we assume the errors are independent. Thus the total errors from the center of area technique may be found by reading the appropriate standard deviation values from Figures 4.21 and 4.24 and taking the square root of the sum of their squares.

The error due to intensity noise increases as the number of samples increases while the error due to sampling offset decreases as the number of samples increases; thus, there is a tradeoff of errors from the two sources. Because the error due to intensity noise will usually be much larger than the errors from sampling offsets, Equation 4.36 and Figure 4.21 should be used for determining errors from the center of area technique.

4.4.6 Match Filtering and Peak Detection

In this approach, the image is smoothed to reduce noise errors then the peak of the intensity profile is found. This approach is similar to edge and feature finding techniques [41,40,121,92,49,118]. It differs from the center of area calculation in that after the smoothing, the center finding is done locally and is not very sensitive to the intensities near the "tails" of profiles.

The subsequent analysis follows Canny's feature detection results [41,40]. We assume the waveform containing the feature is $F(x)$ and the impulse response of the filter (the feature detection function) is $f(x)$. The response of the feature detector at the "center" of the waveform is defined by

$$O(x) = \int_{-\infty}^{\infty} F(-x) f(x) dx \tag{4.38}$$

which is the convolution integral evaluated at the origin. Peak detection of $O(x)$ will give an approximation to the location of the feature x_0 (the peak of the profile with the noise removed). Canny defines the localization, Λ, as the inverse of the approximation of the standard deviation in finding the peak

$$\Lambda = \frac{1}{\sigma_{x_0}} = \frac{\int_{-\infty}^{\infty} F(-x) f''(x) dx}{n_0 \sqrt{\int_{-\infty}^{\infty} f'^2(x) dx}} \tag{4.39}$$

where n_0 is the variance of the noise (assumed Gaussian). Assuming that the waveform is symmetric $[F(-x) = F(x)]$ we may integrate the numerator of Equation (4.39) by parts to obtain

$$\Lambda = \frac{\int_{-\infty}^{\infty} F'(x) f'(x) dx}{n_0 \sqrt{\int_{-\infty}^{\infty} f'^2(x) dx}}. \tag{4.40}$$

We would like to maximize Λ (as well as maximize signal to noise ratio). Using the Cauchy-Schwartz inequality [105] we know that

$$\int_{-\infty}^{\infty} F'(x)f'(x)dx \le \left[\int_{-\infty}^{\infty} F'^2(x)dx\right]^{\frac{1}{2}} \left[\int_{-\infty}^{\infty} f'^2(x)dx\right]^{\frac{1}{2}}.$$

Thus to maximize Λ (as well as the signal-to-noise ratio [41,40]), we want $f'(x) = F'(x)$ (the "match" filter). Using the match filter in Equation (4.39), the standard deviation in locating the feature is

$$\sigma_{x_0} = \frac{n_0}{\sqrt{\int_{-\infty}^{\infty} f'^2(x)dx}}. \qquad (4.41)$$

Assuming the intensity profile is approximately Gaussian [Equation (4.28)], the standard deviation in locating the stripe is

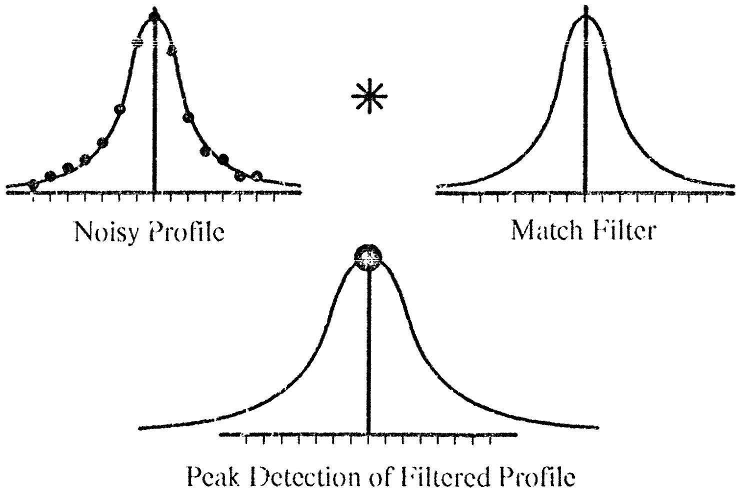
$$\sigma_{x_0} = \frac{\sqrt{2\sigma}}{\sqrt[4]{\pi}}\left(\frac{n_0}{A_{max}}\right). \qquad (4.42)$$

Equation (4.42) gives the standard deviation in locating the peak of the light stripe as a function of the ratio of the noise standard deviation to the maximum intensity amplitude and the square root of the width of the stripe intensity. Thus, the noise may be minimized by minimizing the width of the light stripe; however, sampling too thin a stripe will give rise to undersampling problems (see Section 4.4.2). For a noise-to-maximum-intensity ratio of five percent, match filtering the finite width stripe gives a result is about four times more accurate ($\frac{1}{4}$ of the standard deviation) than the results for the thin line.

In summary, the match-filter/peak-detection technique filters the noisy intensity profile with a similarly shaped filter, Figure 4.25. The filtered profile is then scanned for the peak whose location is an approximation to the location of the peak of the profile with the noise removed.

4.5 Using Redundant Sensed Information

Techniques for combining similar information from a number of sources are now considered. The techniques discussed are applicable to many types of sensed information; in fact, fusion of information from fundamentally different sources is possible as long as a measure of the expected accuracy (covariance matrix) of the information is available. The analysis in this section closely follows the derivation in [64].


Noisy Profile

Match Filter

Peak Detection of Filtered Profile

Figure 4.25: The match-filter and peak detection procedure for determining the location of the peak of an intensity profile entails filtering the image with a match filter then performing a peak detection in the filtered profile.

4.5.1 Optimal Estimation Theory

Optimal estimation theory may be used to find the best estimate for the orientation of a feature from redundant sensed information. Measurements, $\vec{z}$, may be expressed as a function of the actual states, $\vec{x}$, and measurement noise, $\vec{v}$, [64]

$$\vec{z} = \mathbf{H}\vec{x} + \vec{v} \tag{4.43}$$

where $\mathbf{H}$ is the matrix which relates the states to the measurements. To find an estimate, $\hat{x}$, for the states, $\vec{x}$, we may use a maximum likelihood philosophy or a Bayesian approach. In the maximum likelihood approach, no *a priori* knowledge about the actual states, $\vec{x}$, is assumed and the probability of obtaining the measurements, $\vec{z}$, from some state, $\hat{x}$ is maximized. That is, we wish to find the $\vec{x}$ which maximizes $p(\vec{z}|\vec{x})$. In the Bayesian approach a prior distribution on $\vec{x}$ is assumed. This will be altered by new measurements $\vec{z}$. Depending on the criterion for optimality, we may compute an estimate of the states from Bayes' theorem

$$p(\vec{x}|\vec{z}) = \frac{p(\vec{z}|\vec{x})p(\vec{x})}{p(\vec{z})} \tag{4.44}$$

where $p(\vec{z})$ is the probability density function of the measurements.

Optimal Estimates using a Maximum Likelihood Philosophy

The conditional probability density function for $\vec{z}$ conditioned on $\vec{x}$ has the identical shape as the probability density function for $\vec{v}$, but has a mean of $\mathbf{H}\vec{x}$. With the assumption that $\vec{v}$ is a zero mean, gaussian distributed random variable with covariance matrix **C**, we may obtain an expression for $p(\vec{z}|\vec{x})$

$$p(\vec{z}|\vec{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{C}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\vec{z}-\mathbf{H}\vec{x})^T\mathbf{C}^{-1}(\vec{z}-\mathbf{H}\vec{x})\right] \tag{4.45}$$

where n is the order of $\vec{z}$. To maximize Equation (4.45) we must minimize the exponent. Differentiating the exponent with respect to $\vec{x}$ and setting this equal to zero, we obtain a value for $\vec{x}$ which is the maximum likelihood estimate

$$\hat{x} = (\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\vec{z} \tag{4.46}$$

Optimal Estimates Using a Bayesian Philosophy

A Bayesian approach to the optimal estimation problem assumes that there is some prior knowledge about the states $\vec{x}$ represented by the prior density function $p(\vec{x})$; that is, we assume a prior mean, $\hat{x}^-$ and a prior covariance matrix $\mathbf{P}^-$. If we choose a minimum variance optimality criterion, the $\vec{x}$ which minimizes the criterion is the conditional mean of $\vec{x}$

$$\hat{x} = E[\vec{x}|\vec{z}].$$

We may calculate the posterior state estimate, $\hat{x}^+$ by assuming gaussian distributions for $\vec{x}$ and $\vec{v}$

$$\hat{x}^+ = \hat{x}^- + \mathbf{K}[\vec{z} - \mathbf{H}\hat{x}^-] \tag{4.47}$$

where **K** is the Kalman gain matrix defined as

$$\mathbf{K} = \mathbf{P}^-\mathbf{H}^T[\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{C}]^{-1} \tag{4.48}$$

and **C** is the covariance matrix for the measurement vector $\vec{z}$. The covariance matrix for the new estimate $\hat{x}^+$ is

$$\mathbf{P}^+ = [\mathbf{I} - \mathbf{K}\mathbf{H}]\mathbf{P}^- \tag{4.49}$$

where **I** is the identity matrix.

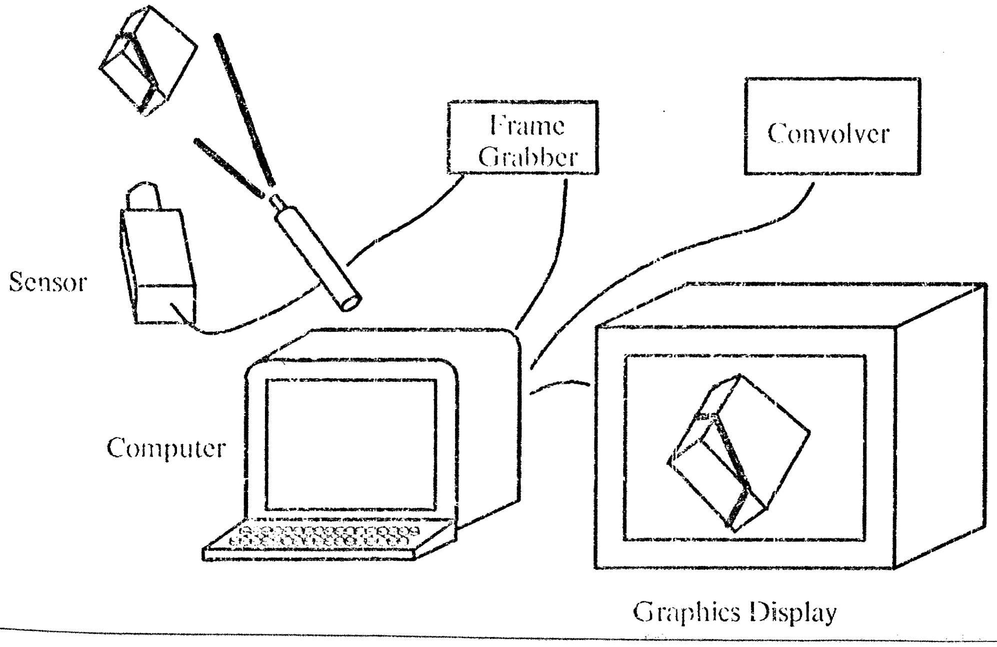


Figure 4.26: Sensing and image processing system components.

Equation(4.47) gives a recursive estimation technique for determining the estimate of a number of parameters from sensed data. The noise from the consecutive measurements must be independent ($\vec{z}$'s are independent). The measurements, $\vec{z}$, need not be from the same source. For instance, measurement estimates of a part's orientation may come from gripper positions, prior knowledge, and part position sensor output. Each of these sources provide some additional information to the estimate and are weighted according to their expected accuracy.

4.6 Processing Light Stripe Images

4.6.1 Sensing and Image Processing Hardware

The sensing and image processing system is shown schematically in Figure 4.26. It consists of a laser and cylindrical lens assembly, a video camera, a controlling computer, a frame grabber, a hardware convolver, and an optional color graphics display.

4.6.2 Image Processing Steps

An object is initially located in the sensor area. The sensor area is defined to be the area of the light plane which would be visible to the camera were the plane an opaque solid sheet. The camera, which is externally synced by the computer, is constantly sending video information to the frame grabber. When so instructed by the controlling computer, the frame grabber records a frame from the camera. The information in the frame grabber array, which is essentially an extension of the computer's memory, is then analyzed by the computer. The computer crops a smaller array which contains the features of interest (the curve generated by the visible intersection of the light plane and part surfaces) out of the image array. The cropped array is then passed through the hardware convolver where it is smoothed with a Gaussian filter. The filtered image of the intersection curve is then segmented into separate regions to which curve fitting algorithms may be applied. Each row is then scanned to determine the location of the peak intensity of the light stripe (approximately the center of the profile), then the curve fitting algorithms are applied to each segment of the intersection curve. The fitted curve parameters along with the calibrated sensor model parameters are then used to determine the location of the feature in space.

In the following description of details of the image processing system, we assume that each segment of the intersection curve is nearly a straight line. That is, all surfaces of interest of the part are planar. This constraint limits the types of parts which may be sensed to ones with polyhedral features.

Filtering

A two-dimensional convolution operation is used to smooth the noise in the image of the intersection curve. This operation improves the accuracy with which the "center" of the light stripe may be found. Metrics for determining the location of light stripe features are discussed in Section 4.4. Because a design specification for the sensor is real time operation, it is necessary to use a hardware convolver for this image processing step.

If the line segments in the image are nearly vertical (this would be the case for many segments generated from a laser plane which is nearly vertical) filtering in one dimension along the rows of the image would be sufficient. This type of filtering may be performed by a filter operating on the video signal directly and would appreciably speed this processing step.

Segmentation

The intersection curve of a light plane and a polyhedral object consists of a number of straight line segments linked end to end. In order to determine the parameters of each of the segments it is necessary to determine where one segment ends and the other starts. In addition, not all real parts have surface bounding edges with high curvature (sharp corners) and not all lines in the image correspond to light plane intersections with faces of interest; thus, the segmentation algorithm must be robust enough to ignore unwanted information. Curve segmentation algorithms have been extensively reported in the pattern recognition, computer graphics, and signal processing literature (for example see [150,157] or Reference [3] for an overview).

A few of these algorithms were implemented. A merging algorithm was initially used. It essentially walks along the original curve from one end to the other and checks whether a new data point is out-of-line with a line fitted to the previous data points. If the distance to the new point is above a certain threshold, the algorithm splits the data and starts checking along a new line.

A recursive split-and-merge algorithm similar to the one reported by Ramer [157] was also used. For simplification, we assume that none of the line segments are nearly horizontal (this is not a bad assumption for a system with a near vertical light plane source). The splitting phase uses the following steps (see Figure 4.27).

1. Construct a straight line between the endpoints of the curve to be segmented. Call these end points P_1 and P_2.

2. Find the point on the light stripe along the same row as the midpoint of the constructed line. Call this the test point, P_T.

3. Calculate the perpendicular distance from the constructed line to P_T.

4. If the distance is within some threshold level, exit returning the constructed line.

5. Otherwise, split the line and recursively apply step 1 to two sets of new end-points: (P_1, P_T) and (P_T, P_2).

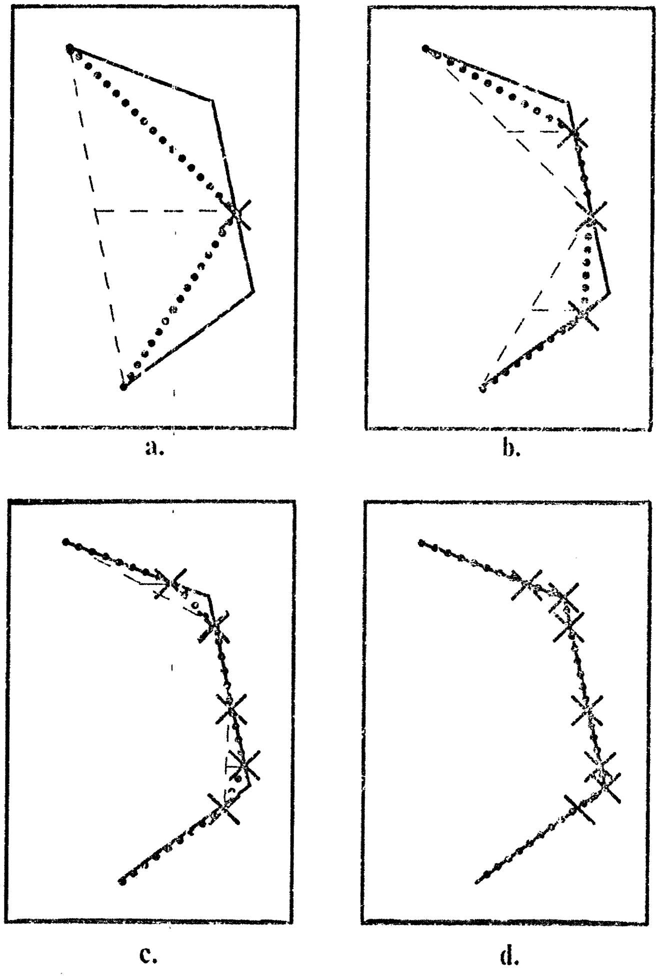


Figure 4.27: Recursive split-and-merge segmentation algorithm. The center of a line drawn between the two end points in an image of line segments is tested for being near the line segments. If it is not, the proposed line is split and the algorithm is applied to the two new lines.

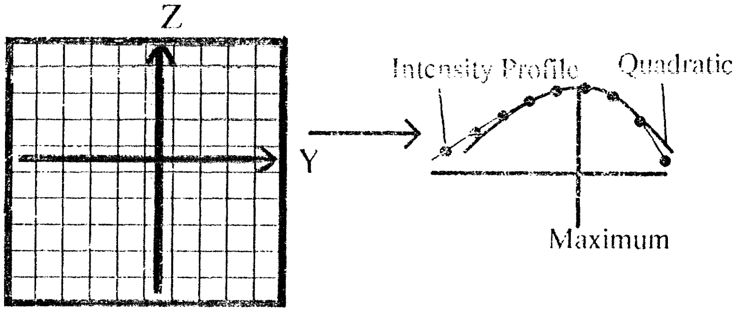


Figure 4.28: Coordinates used for the line fitting algorithm.

Generally, after the splitting is complete, a number of groups of points lie along the same line, Figure 4.27d. A merging algorithm is then used to consolidate the point groups.

1. The slope of a group of points is determined from the positions of the end points. The relative angle between adjacent groups is also calculated.
2. If the angle between adjacent groups is above some threshold, the breakpoint between the groups is maintained.
3. Otherwise, the groups are merged in to one group.
4. The algorithm starting with step 1 is iterated until no further merging occurs.

The split-and-merge procedure was found to be more robust than the merge algorithm when dealing with images with smooth transitions between line segments. The computation times required for the two algorithms were similar.

Peak Detection

The coordinates used in the line fitting procedure are shown in Figure 4.28. One data point is taken for each row of the image (z coordinate). The y coordinate is taken to be the peak of the filtered image along that row. The location of the peak is found to a subpixel level by finding the maximum of a quadratic fitted to the points near the peak intensity along that row. The maximum intensity in the first row is found by scanning the entire row. The maximum intensities in remaining rows are found by locally scanning each row near the y coordinate of the maximum

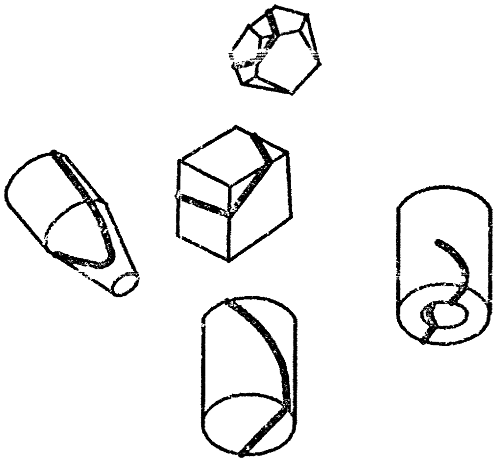


Figure 4.29: Some features which may be located in five or six degrees of freedom from data generated by the intersection of a single light plane.

from the previous row. Thus, only a small percentage of the pixels in the image are actually scanned.

Linear Least Squares Line Fitting

After segmentation, filtering, and peak detection, the best fit line parameters must be determined for the sets of data points. In the least squares technique, the sum of the squares of distances from the best-fit line to the data points are minimized. Errors parallel to the pixel coordinates are minimized (see Section 4.3 for a discussion of the technique and choice of metric to minimize).

4.7 Measuring the Location of Features with a Single Light Stripe

With the right geometric relationship, the intersection of a light plane and certain geometric features will generate curves of intersection which contain enough information to locate the features in six degrees of freedom (or five degrees of freedom for rotationally symmetric parts), Figure 4.29. The following analysis deals mostly with polyhedral features. Also, we assume a known correspondence between the

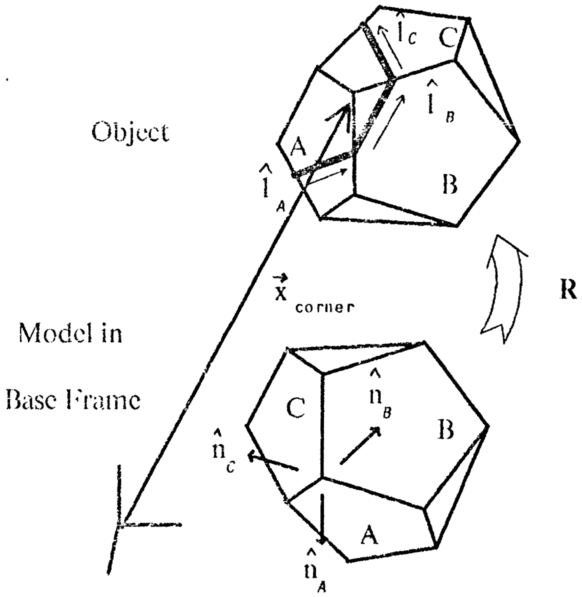


Figure 4.30: Light plane intersecting a polyhedral feature and a corresponding model of the feature.

groups of data from the sensor and the faces of the model (for instance, we know which surface on the part generated data comprising a particular line segment).

4.7.1 Locating General Polyhedral Features

This section discusses the calculations involved in determining the orientation and displacement of a polyhedral feature of an object. The position is referenced relative to a model of the feature in a base coordinate system, Figure 4.30. The orientation of the feature may be determined by the slopes of three line segments in the image. The displacement may be determined from the results of the orientation calculation and the intercepts of the three lines; thus, the orientation calculation may be decoupled from the displacement calculation.

Sufficient Conditions for Position Constraint

The sufficient conditions for three lines lying on three faces of a polyhedron to constrain the polyhedron in six degrees of freedom may be determined using screw theory [16,143,164]. Each line segment provides constraint in two freedoms; translation perpendicular to the surface and rotation about an axis perpendicular to both the normal of the surface and the line. With the proper geometric conditions, three line segments will provide six freedoms of constraint. According to Salisbury [164], these conditions are

- Pairs of lines must not be parallel.
- Pairs of faces in which lines lie must not be parallel.
- The common normal of any pair of lines must not be parallel to the line of intersection of any two faces in which the lines lie.

Determining the Orientation

Assume that the three normal vectors for faces A, B, and C of the model in Figure 4.30 are known to be $\hat{n}_A$, $\hat{n}_B$, and $\hat{n}_C$ respectively. The normal vectors of the object are unknown, but vectors along the intersection line segments are known from the sensed data. Assuming that the correspondence between faces is known, the unit vectors along the intersection line segments for faces A, B, and C are $\hat{l}_A$, $\hat{l}_B$, and $\hat{l}_C$ respectively. The rotation matrix $\mathbf{R}$ from the model to the object is found using quaternion notation [93,151,163]. We denote a quaternion by the bold face letter, $\mathbf{q}$. Every quaternion has a corresponding four vector, $\vec{q}$, whose components are the four elements of the quaternion. A quaternion with a zero scalar element represents a three vector; thus a quaternion multiplied by a vector is defined. A rigid rotation of a vector $\vec{v}$ is

$$\mathbf{q}\vec{v}\mathbf{q}^*$$

where $\mathbf{q}$ is a unit quaternion and $\mathbf{q}^*$ is the quaternion complement of $\mathbf{q}$.

If the proper rotation is found, the line vectors, $\hat{l}_i$, will be perpendicular to the rotated model normal vectors, $\mathbf{q}\hat{n}_i\mathbf{q}^*$; that is, the dot product will be zero

$$\hat{l}_i \cdot \mathbf{q}\hat{n}_i\mathbf{q}^* = 0, \quad i = A, B, C. \tag{4.50}$$

Each of these equations are quadratic in components of $\mathbf{q}$. Rewriting this in matrix form

$$\begin{aligned} \vec{q}^T \mathbf{A}_i \vec{q} &= 0, \quad i = A, B, C, \\ |\vec{q}| &= 1 \end{aligned} \tag{4.51}$$

which is a coupled set of four quadratic equations. We can solve for the components of $\vec{q}$ using a numerical technique. The rotation matrix, $\mathbf{R}$, may be calculated from the components of $\mathbf{q}$ [163]

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(-q_0q_3 + q_1q_2) & 2(q_0q_2 + q_1q_3) \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(-q_0q_1 + q_2q_3) \\ 2(-q_0q_2 + q_1q_3) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Determining the Displacement

Once the rotation from the model to the object is known, the normal vectors of the object, $\hat{n}_{obj,i}$, may be calculated

$$\hat{n}_{obj,i} = \mathbf{R}\hat{n}_i, \quad i = A, B, C. \tag{4.52}$$

The locations of a line on the i^{th} face is defined by the intercept with the plane $z = 0$, We denote the intercept vector $\vec{b}_i$. The equation of the plane which lies on the i^{th} face is [61]

$$(\vec{x} - \vec{b}_i) \cdot \hat{n}_{obj,i} = 0. \tag{4.53}$$

where $\vec{x} = [x\ y\ z]^T$ is the location of a point in space. Solving these three equations simultaneously for the planes on faces A, B, and C, we obtain the location of the corner [61]

$$\vec{x}_{corner} = \frac{p_A(\hat{n}_{obj,B} \times \hat{n}_{obj,C}) + p_B(\hat{n}_{obj,C} \times \hat{n}_{obj,A}) + p_C(\hat{n}_{obj,A} \times \hat{n}_{obj,B})}{\hat{n}_{obj,A} \cdot (\hat{n}_{obj,B} \times \hat{n}_{obj,C})} \tag{4.54}$$

where p_i is the perpendicular distance from the origin to the i^{th} plane

$$\begin{aligned} p_A &= (b_A \cdot \hat{n}_{obj,A}) \\ p_B &= (b_B \cdot \hat{n}_{obj,B}) \\ p_C &= (b_C \cdot \hat{n}_{obj,C}). \end{aligned} \tag{4.55}$$

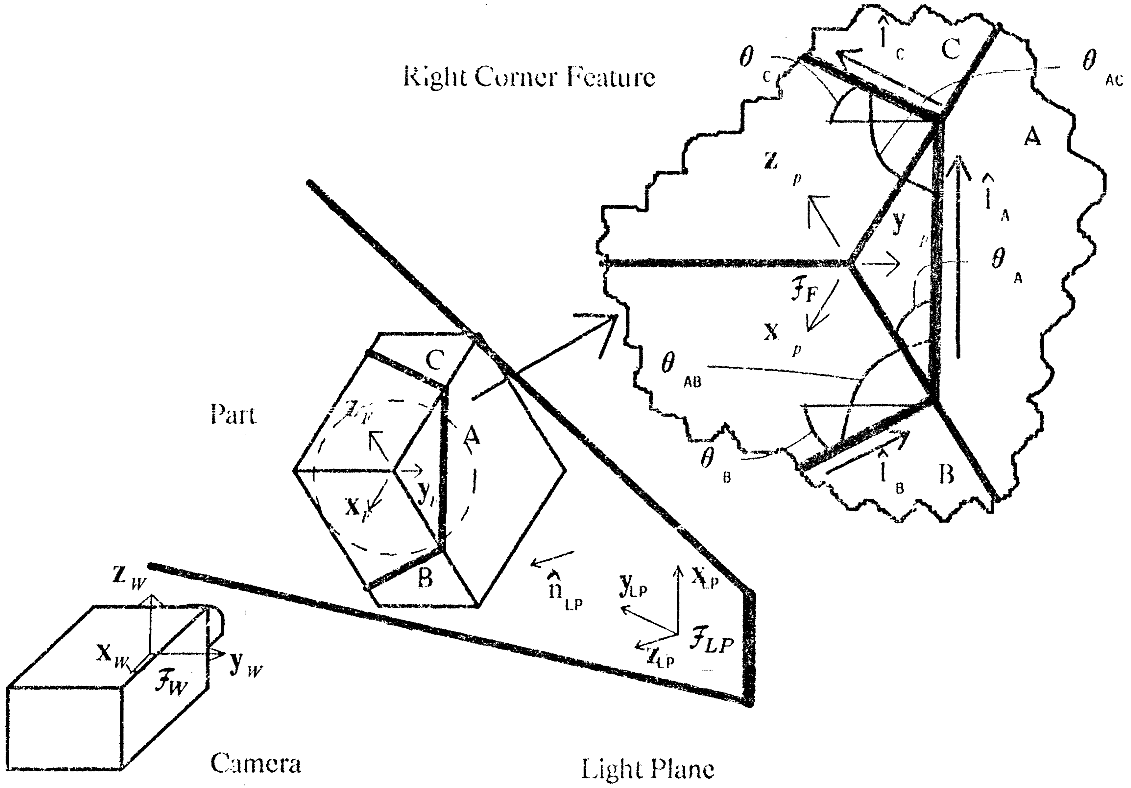


Figure 4.31: A right corner feature and reference frames used to determine its orientation.

4.7.2 Locating Right Corner Features

In this section, the solution for the rotation of the right polyhedral corner feature is solved in closed form (in contrast to using a numerical technique for general polyhedra) using a direct method of calculation.

We define a right corner feature as the intersection of three half spaces (see Figure 4.31) whose bounding planes are mutually orthogonal. A right-hand coordinate system, $\mathcal{F}_F$, with axes $\hat{x}_F$, $\hat{y}_F$, $\hat{z}_F$ perpendicular to the bounding planes of the half spaces and origin located at the point of intersection of the bounding planes defines the position of the feature in six degrees of freedom. We wish to determine the orientation and position of this feature reference frame with respect to the world frame, $\mathcal{F}_W$, located at the camera image plane.

Determining the Orientation

The orientation of the feature reference frame, $\mathcal{F}_F$, with respect to the world reference frame, $\mathcal{F}_W$, is determined by locating an intermediate reference frame associated with the laser plane, $\mathcal{F}_{LP}$. For the present analysis, we assume that the light plane has no width (two-dimensional). We also assume that three line segments (no width) are generated when the light plane intersects the feature. The orientation of the three line segments are defined by the three vectors $\hat{l}_A$, $\hat{l}_B$, and $\hat{l}_C$ as shown in Figure 4.31.

We assume the ranging sensor has been accurately calibrated and the unit vector normal to the light plane, $\hat{n}_{LP}$, is known. The intermediate coordinate system, $\mathcal{F}_{LP}$, has axes $\hat{x}_{LP}$ in direction of one of the lines of intersection $(\hat{l}_A)$, $\hat{z}_{LP}$ in direction $\hat{n}_{LP}$, and $\hat{y}_{LP}$ in direction $\hat{n}_{LP} \times \hat{l}_A$ where × is the vector product.

The rotational transformation (in matrix form) from $\mathcal{F}_W$ to $\mathcal{F}_{LP}$ is

$$\mathbf{R}_{W,LP} = \left[\hat{l}_A,\ \ \hat{n}_{LP} \times \hat{l}_A,\ \ \hat{n}_{LP}\right] \tag{4.56}$$

$\mathcal{F}_{LP}$ may be located with respect to $\mathcal{F}_F$ by realizing that the cross product of any two of the three vectors $\hat{l}_A$, $\hat{l}_B$, or $\hat{l}_C$ is normal to the light plane (in the direction of $\hat{z}_{LP}$) and $\hat{l}_A$ lies on the $\hat{y}_F, \hat{z}_F$ plane. The transformation from $\mathcal{F}_W$ to $\mathcal{F}_{LP}$ is[3]

$$\begin{aligned}
\mathbf{R}_{F,LP} &= \left[{}^F\hat{l}_A,\ \ ({}^F\hat{l}_B \times {}^F\hat{l}_A) \times {}^F\hat{l}_A,\ \ {}^F\hat{l}_B \times {}^F\hat{l}_A\right] \qquad (4.57)\\
&= \begin{bmatrix}
0 & \frac{\sin\theta_B}{\sqrt{\cos^2\theta_A \sin^2\theta_B + \sin^2\theta_A}} & \frac{\sin\theta_A \cos\theta_B}{\sqrt{\cos^2\theta_A \sin^2\theta_B + \sin^2\theta_A}} \\
-\cos\theta_A & \frac{\sin^2\theta_A \cos\theta_B}{\sqrt{\cos^2\theta_A \sin^2\theta_B + \sin^2\theta_A}} & \frac{-\sin\theta_B \sin\theta_A}{\sqrt{\cos^2\theta_A \sin^2\theta_B + \sin^2\theta_A}} \\
-\sin\theta_A & \frac{-\cos\theta_B \cos\theta_B \sin\theta_A}{\sqrt{\cos^2\theta_A \sin^2\theta_B + \sin^2\theta_A}} & \frac{\sin\theta_B \cos\theta_A}{\sqrt{\cos^2\theta_A \sin^2\theta_B + \sin^2\theta_A}}
\end{bmatrix}
\end{aligned}$$

where θ_A is the angle between $\hat{l}_A$ and $\hat{z}_F$ and θ_B is the angle between $\hat{l}_B$ and y_F. The angles θ_A and θ_B may be calculated from the angle between $\hat{l}_A$ and $\hat{l}_C$ (θ_{AC}) and the angle between $\hat{l}_A$ and $\hat{l}_B$ (θ_{AB}) (these angles are both easily obtainable from the processed image).

$$\theta_A = \arctan\sqrt{-\frac{\tan\theta_{AB}}{\tan\theta_{AC}}}$$

[3] The left superscripts ${}^F\hat{l}$ denotes that the vector is expressed with respect to the $\mathcal{F}_F$ reference frame.

$$\theta_B = \arccos \frac{\cos \theta_{AB}}{\cos \theta_A}$$

Multiple values of the rotation matrix, $\mathbf{R}_{F,LP}$, which correspond to different light stripe image interpretations result from these equations. The proper rotation matrix is the one closest to the expected nominal rotation matrix. The orientation of the feature reference frame, $\mathcal{F}_F$, with respect to the world coordinate frame, $\mathcal{F}_W$, is

$$\mathbf{R}_{W,F} = \mathbf{R}_{0,LP}\mathbf{R}_{F,LP}^{-1}. \tag{4.58}$$

Determining the Displacement

Once the rotation is known, the displacement of the corner from the base coordinate system is given by Equation 4.54.

4.7.3 Locating Other Features

In order to extend the range of parts which may be handled by the sensor, features other than just polyhedral ones should be measurable. Locating parts with quadric surfaces (cylinders, cones, spheres, etc.) was considered, but not implemented in detail. The intersection of the light plane and a quadric surface generates conic sections. In general, the perspective projection of a conic changes its parameters; thus, curve fitting should be done in the light plane (that is, points recorded in image plane coordinates should be projected back to the light plane). Algorithms which fit conic sections are given in [6,30,166]. A number of the algorithms were tried on synthetic quantized data of the intersection curve between a light plane and a cylindrical feature. As noted in [166] the Bookstein algorithm [30] was the most accurate of the conic fitting routines; although, it was significantly less accurate in locating a feature than the right corner algorithm. Troubles in using ellipse fitting routines on light stripe data were reported by Bolles and Fischler [26]. In their RANSAC approach, they chose a set of five points, constructed an ellipse and tested to see if it was a reasonable estimate. If it is not reasonable, the procedure is tried again.

4.8 Using Multiple Light Planes to Locate Polyhedral Features

With the sufficient conditions outlined in Section 4.7, three lines locate a polyhedron in six degrees of freedom; thus, more than three lines can overconstrain the calculation. In this section, a least squares approach is used to solve the overconstrained problem. A technique which performs a least squares estimate can more precisely determine a feature's position than a technique which just uses the minimum number of line segments.

The equivalent of four or more line segments may be generated on a polyhedral feature either by using two or more images with the part or light plane displaced slightly, or by using multiple light planes and a single image. In addition to allowing a more accurate calculation of the location of a part feature, a system of multiple light planes has the advantage of having illumination sources which are not in a single plane; thus, it may be possible to generate longer line segments or have line segments in more desirable locations with respect to the part feature. In fact, some features may be localized with a number of light planes but not with a single light plane.

4.8.1 Non-Optimal Orientation Estimation for Polyhedral Features

This technique is non-optimal because it doesn't take the quality of different measurements into account. It is, however, somewhat simpler than the optimal estimate and is similar to approaches taken in the literature [59,60,58]. A least squares technique is used to determine the best fit rotation for the general polyhedral object. Because no surface normal information is directly available in the sensed data, we cannot use the technique developed by Faugeras et al. [59,60,58]. The approach and the notation used here is similar to that in Section 4.7.1. The sum of the squares of the dot products are minimized

$$min \ \sum_i (\hat{l}_i \cdot \mathbf{q}\hat{n}_i\mathbf{q}^*)^2. \tag{4.59}$$

A value for $\mathbf{q}$ which minimizes this expression may be found using a numerical technique.

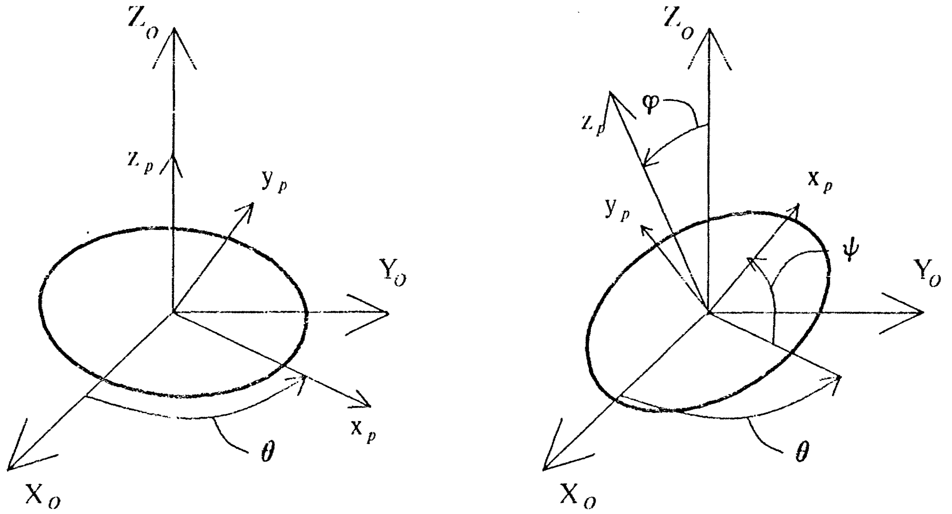


Figure 4.32: Euler angles used in the least squares estimate for the orientation of a polyhedral feature.

4.8.2 Optimal Estimation of Orientation for a Polyhedral Feature

Representation of Rotations

Three independent rotations and three independent displacements are sufficient to describe the position of an object in three-space. A number of different descriptions are available to model the three rotational degrees of freedom.[4] For the subsequent analysis we use the euler angle description because the number of variables needed to describe the rotation is equal to the number of degrees of freedom. The euler angle convention is shown in Figure 4.32. A coordinate frame located at euler angles

$$\vec{\alpha} = \left\{ \begin{matrix} \theta \\ \phi \\ \psi \end{matrix} \right\}$$

is first rotated about the base system z axis by angle θ then about its own x axis by angle ϕ, then about its own z axis by angle ψ.

We assume that orientation estimates expressed in euler angle notation, $\{\vec{\alpha}_i\}$,

[4] Frequently used descriptions for rotations are the 3 × 3 orthonormal rotation matrix [93,147], euler angles [93,47], and quaternions [93,151,163].

are available. In general, these may be obtained from three appropriately chosen line segments.

$$\vec{\alpha}_i = \vec{f}_i(S_{i,1}, S_{i,2}, S_{i,3}), \quad i = 1, \ldots, N \tag{4.60}$$

where $\vec{f}_i$ is a vector function, $S_{i,1}, S_{i,2},$ and$S_{i,3}$ are the slopes of the three line segments from the sensor, and N is the number of different estimates. We wish to combine these estimates to arrive at the most accurate estimate possible, $\hat{\alpha}$, given the expected errors of the measurements.

The functions $\vec{f}_i$ are functions of the calibration variables and part model parameters as well as the slopes of the line segments. For simplicity, only the errors in slope are considered in the following Kalman filter design. If the magnitude of the unmodeled uncertainties are large enough, imprecise results may be generated.

In general the functions $\vec{f}_i$ in Equation (4.60) are non-linear and we cannot directly apply linear optimal estimation theory with the state vector, $\vec{x}$, being the euler angles and the measurement vector, $\vec{z}$, being the line slopes. We can, however, assume that errors in slope measurements, $\delta S_{i,\ell_l}$, are small and linearize Equation (4.60). Taking the Taylor series expansion and keeping terms of first order or less

$$\vec{\alpha}_i \approx \vec{\bar{\alpha}}_i + \sum_l \frac{\partial \vec{f}_i}{\partial S_{i,\ell_l}} \delta S_{i,\ell_l} \tag{4.61}$$

where l takes on the appropriate values for the lines used in the estimates and $\delta S_{i,\ell_l}$ is a random variable which is the difference between the actual slope for line l and the measured slope. In matrix form, Equation (4.61) is

$$\vec{\alpha}_i \approx \vec{\bar{\alpha}}_i + \mathbf{K}_{S,i} \delta \vec{S}_i \tag{4.62}$$

where

$$\mathbf{K}_{S,i} = \left[\frac{\partial \vec{f}_i}{\partial S_{i,1}}, \frac{\partial \vec{f}_i}{\partial S_{i,2}}, \frac{\partial \vec{f}_i}{\partial S_{i,3}} \right] \quad \text{and} \quad \delta \vec{S}_i = \left\{ \begin{array}{c} \delta S_{i,1} \\ \delta S_{i,2} \\ \delta S_{i,3} \end{array} \right\}. \tag{4.63}$$

The partial derivatives may be calculated directly if an analytical expression is available for $\vec{f}_i$. If $\vec{f}_i$ involves a numerical technique, the partial derivatives may be approximated by

$$\frac{\partial \vec{f}_i}{\partial S_{i,j}} \approx \frac{\vec{f}_i(S_{i,1}, \ldots, (S_{i,j} + \epsilon), \ldots, S_{i,3})}{\epsilon}$$

where ϵ is a small number whose addition to $S_{i,j}$ will produce a result in the numerical calculation significantly larger than the numerical error.

Rearranging terms, Equation (4.62) becomes

$$\mathbf{K}_{S,i}^{-1}\vec{\bar{\alpha}}_i = \mathbf{K}_{S,i}^{-1}\vec{\alpha}_i - \delta\vec{S}_i. \tag{4.64}$$

All N equations in (4.64) may be combined into a single equation by adding the rows of each vector term corresponding to the same line segment

$$\vec{z}_S = \mathbf{H}_S\vec{\alpha} + \vec{v}_S \tag{4.65}$$

where

$$\vec{z}_S = \sum_i \mathbf{K}_{S,i}^{-1}\vec{\bar{\alpha}}_i, \quad \mathbf{H}_S = \sum_i \mathbf{K}_{S,i}^{-1},$$
$$\vec{v}_S = -\sum_i \delta\vec{S}_i.$$

Equation (4.65) has the same number of rows as independent line slopes in the image.

Equation (4.65) is in the same form as the linear optimal estimation equation (4.43); thus, the maximum likelihood estimate, Equation (4.46), of the euler angles is

$$\hat{\alpha} = (\mathbf{H}_S^T\mathbf{C}_S^{-1}\mathbf{H}_S)^{-1}\mathbf{H}_S^T\mathbf{C}_S^{-1}\vec{z}_S \tag{4.66}$$

where $\mathbf{C}_S$ is the covariance matrix of the noise vector $\vec{v}_S$. The minimum variance Bayesian estimate of the euler angles given an *a priori* estimate $\vec{x}-$ and initial covariance matrix $\mathbf{P}^-$ is [64]

$$\hat{x} = \vec{x}^-([\mathbf{P}^-]^{-1} + \mathbf{H}_S^T\mathbf{C}_S^{-1}\mathbf{H}_S)^{-1}\mathbf{H}_S^T\mathbf{C}_S^{-1}(\vec{z}_S - \mathbf{H}_S\vec{x}^-). \tag{4.67}$$

Assembly Using the A Priori Error Suppression Technique

Chapter 5

5.1 Introduction

In order to perform assembly operations without position sensing, it is necessary to control the propagation of part position errors. Fixtures, pallets, and grippers must be designed to geometrically constrain the parts. The design of these items becomes difficult if many different shaped parts must be handled and fixture interchange is undesirable. This chapter discusses some tools which were developed to address some of the more difficult assembly operations cited in the assembly task analysis in Section 2.2.1. The tools are designed so as to minimize the number of fixture changes during an assembly process. Excessive interchange of fixtures leads to inefficient use of the assembly machine. In the Draper Labs automobile alternator assembly system, about a third of the cycle time was spend interchanging robot end effectors [43].

The assembly tools are integrated with an industrial robot and a manipulator path generation system in a prototype assembly cell. The path generation system automatically calculates via points to relieve the user from manually digitizing an excessive number of robot positions. The manipulator is able to set up the cell by changing grippers and fixtures and is able to assemble and test consumer hand drills.

5.2 Versatile Tools for Programmable Assembly Systems

A number of the more "difficult" assembly operations discussed in Section 2.2.1 may be accomplished using precision assembly techniques and programmable mechanical tooling. Precision assembly entails the mating of parts using only position controlled moves (no sensing or force control). This section describes the elements of a prototype flexible assembly cell which was developed to demonstrate the feasibility of precision assembly techniques and to extend the types of assembly operations possible with a single six degree of freedom mechanical manipulator.

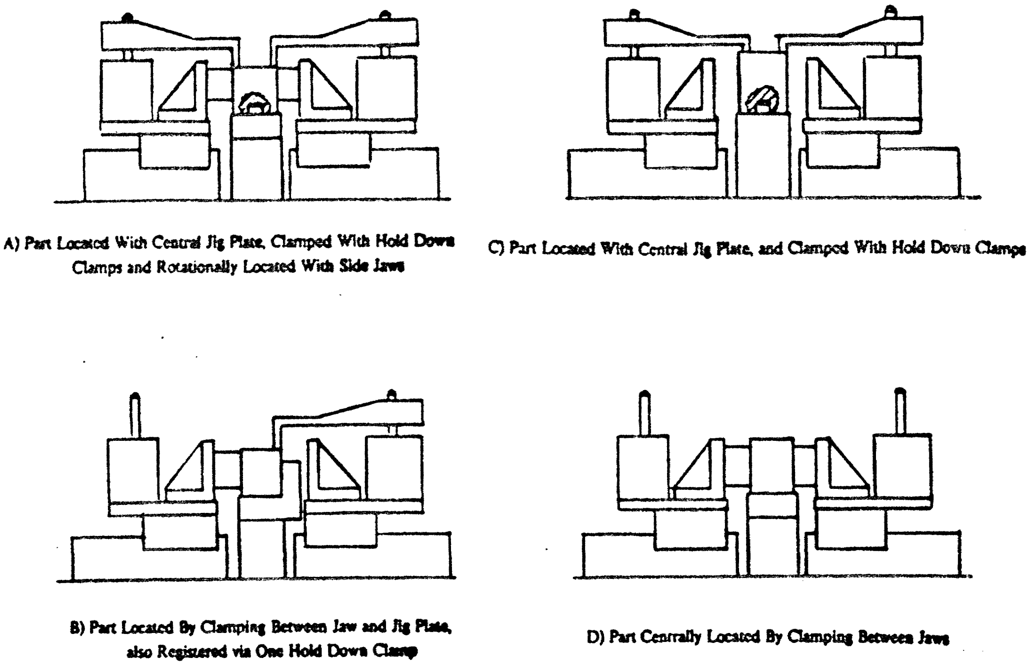
5.2.1 Prototype Assembly Cell Hardware

The prototype assembly cell is a collection of assembly tools which augments the capabilities of a six degree of freedom robot in performing a variety of assembly tasks. A reference base plate with an accurately machined array of locating holes covers the base of the cell. Flexible fixtures are used to assist the robot in the performance of several assembly operations. An assembly vise is used to fixture the base part of an assembly. Since the simple, parallel-jaw gripper of the chosen manipulator is not capable of sufficiently grasping some parts, a special gripper interface and auxiliary fingers were developed which allow the robot to more firmly grasp a larger number of parts.

Assembly Robot

For a system of automated machines being used to their maximum capability (constantly working), the cost of performing the required tasks is a strong function of the process cycle time and, therefore, of manipulator speed. The speed at which robots perform assembly tasks depends upon robot acceleration, deceleration, top speed, servo loop bandwidth, manipulator vibration settling time, firmness of end effector grip, and limiting speeds when mating parts.

At the time the cell was developed, the choice robot was the IBM 7565. This hydraulically driven manipulator has relatively high speed operation, force and optical sensing capabilities, good repeatability (about 0.1 mm.), and is controlled by the powerful AML language [191]. Because the 7565 has six degrees of freedom, it is facile enough to perform most of the necessary tasks in the study; although, a



A) Part Located With Central Jig Plate, Clamped With Hold Down Clamps and Rotationally Located With Side Jaws

C) Part Located With Central Jig Plate, and Clamped With Hold Down Clamps

B) Part Located By Clamping Between Jaw and Jig Plate, also Registered via One Hold Down Clamp

D) Part Centrally Located By Clamping Between Jaws

Figure 5.3: Different combinations of vise fixturing systems which may be used to constrain the base part in an assembly.

fixtured without relying on the location of (inaccurate) external surfaces. After the part has been located and properly secured by a set of clamps, the entire pin plate can be moved down and then laterally out of the way. These motions are provided by pneumatic cylinders, and are actuated by the 7565 controller. This feature allows another part to be inserted through the locating hole in the base part.

Two jaws close about the base part from the sides. The jaws are kinematically constrained to remain equidistant from the center of the jig plate as they close. These jaws are also specific to the assembly task and may be changed by the robot as part of the vise setup procedure. The jaw carriages are DC-motor driven, and can be locked in position by an electronically actuated brake. Jaw motion is terminated in one of two ways: in the first mode, the jaws run into the base part until the motor stalls, and then the brake is applied; the second manner employs a software timer to provide open-loop position commands. These relatively simple jaw positioning schemes eliminate the need for more expensive and complex servo controls.

The final moving elements of the vise are a pair of hold-down clamps used to secure the base part. The clamps are mounted on the jaw carriages so that they also move about the vise center. A set of clamps forms a turret of tools which can be positioned by the robot over the part to be secured. The clamps are actuated vertically by pneumatic cylinders which are also controlled by the computer.

Flexible Fixtures

The assembly task analysis in Section 2.2.1 demonstrated that a set of actuatable fixtures, positioned around the assembly jig, could greatly increase the capability of a single manipulator arm. Many of the tasks which cannot be performed by a single manipulator require only one or two additional active degrees of freedom; thus, a second fully-programmable six degree of freedom manipulator is unnecessary. The success of the flexible fixture design depends greatly upon the accuracy and programmability of the robot. Each fixture is essentially a passive manipulator with a single active degree of freedom at its end effector. The passive joints may be locked and unlocked on computer command. The active degree of freedom, also triggered by the robot controller, is used to reposition the fixture end effector during the assembly operation.

consists of alternating cylinders and spheres, each with a hole bored through its center. These elements are strung onto a steel cable which when placed under high tension makes the structure rigid. The tension in the cable is controlled by a computer-actuated air cylinder; thus, the fixture may be made rigid or slack as directed by the computer. The end effectors of the fixtures are mounted with special quick-release adapters which are necessary if the robot is to interchange the special tooling.

The fixtures are intended to be used for a number of different tasks. With the appropriate end effectors in place, they assist the robot in performing assembly operations on workpieces which would otherwise be unstable (see Section 2.2.1 for a definition of unstable assemblies). After the robot positions the potentially unstable part onto the assembly, the fixture is actuated and its "finger" momentarily secures the part while the robot performs the stabilizing operation, such as driving a screw, or inserting a shaft. The fixture may also be used as a reference surface for the assembly. For example, as more parts are stacked onto a base part, the uncertainty in absolute position of the last inserted part increases. A flexible fixture may be used to locate an assembly so the robot can proceed with the task. It may also be used to hold a sub-assembly in a certain orientation pending the assembly of the next part. Additionally, when provided with more sophisticated end effectors, the simple flexible fixtures could be used to perform an auxiliary assembly tasks, such as testing or inspection.

A force of about 3 lbs. (1.5 kg.) could be withstood at the end of the fixture's arm before the frictional forces within the arm were overcome. This upper-bound on the force is a limitation of the device in performing many tasks. Since proof-of-concept and short development time were goals in building the prototype system, suboptimal arm strength using off-the-shelf hardware was acceptable. An analysis of arm component size versus arm strength is presented in Appendix C. An appropriate arm for a certain set of tasks may be determined from this analysis.

Prototype Pallets

Extension of part feeding cost research by Redford [159,160] shows that off-line feeding can reduce the cost of programmable feeding if the cost of pallets can be kept relatively low. Off-line feeders are decoupled from the assembly process and are not assembly-machine paced. In order to transfer single parts and sub-

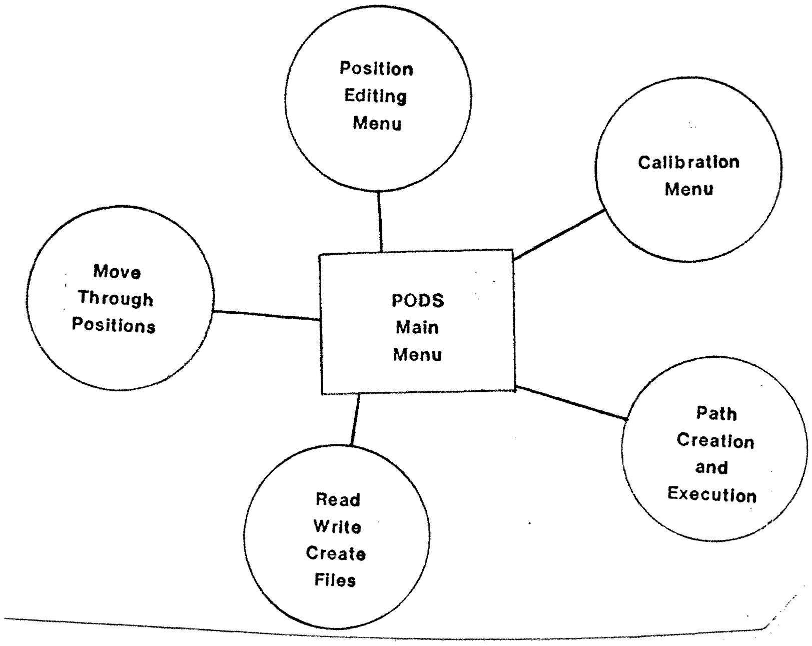


Figure 5.9: Position definition system.

5.3 A Hybrid On/Off-Line Programming System

In order to simplify the current practice of generating robot trajectories using the "teach by showing" technique, a system was developed which allows the programmer to generate a trajectory off-line. Although a number of authors report off-line programming systems [52,107,110,113,116,152,189], the state-of-the-art programming method in the factory is still teach by showing. The technique described here utilizes a relatively small set of general paths or trajectory shapes and path transforms to perform most of the robotic operations necessary in assembly tasks. With this technique, a number of robot configurations must still be manually taught by showing; however, this number is relatively small (usually one position per task). A user friendly Position Definition System (PODS program) was developed to provide an easy method of storing and altering a sequential set of robot configurations.

5.3.1 Position Definition System

A menu driven AML program has been created to help the programmer record and edit robot configurations for use in subsequent assembly programs. This Position Definition System (PODS program) performs a number of functions related to the creation and debugging of position files (see Figure 5.9). A position file is a file containing a list (set of records) of sets of robot joint angles (AML aggregates of 7 real numbers). In addition to the sets of joint angles, a set of reference post locations corresponding the positions of 1 to 4 posts in the workspace are stored as the first record of the file. These posts are used as a reference frame for subsequent playbacks of the positions in the file.

The main menu in PODS allows the user to choose one of the following submenus:

- Choose or Create Position File
- Recalibrate
- Assemble (playback positions)
- Change Speed
- Edit positions
- Create a path
- Execute a path

The user responds to the menu by typing the first two letters of the chosen submenu or ST to stop the program.

The *Choose or Create Position Files* option prompts the user for the name of a position file. PODS then checks to see if a file with such a name exists and if so reads in the reference post locations and the positions. If no file exists with that name, a new file is allocated and the user is automatically shown the calibration menu.

The *Calibration* submenu gives the user a number of choices in the selection of an appropriate set of reference post locations. The user may read in the post locations from another previously defined position file, execute the full calibration procedure thus changing the stored reference post locations, calibrate only the rotary joints, or find a new set of reference posts and use these locations to reference any newly defined points without changing the stored set of post locations (see Section 5.4.2 for a more detailed description of the calibration procedure).

The *Assemble* submenu prompts the user for starting and ending position numbers and then moves the robot through the commanded set of positions. A position number is an index within a sequence of positions listed in a file. The robot speed may be changed using the change speed submenu.

The *Position Editing* submenu allows the user to perform the following editing operations on a set of positions:

- Insert positions using guide box
- Manually insert positions from the keyboard
- Change positions using guide box
- Manually change positions from the keyboard
- Copy positions
- Delete positions
- Display positions
- Print positions
- Choose or create position file

The *Insert* commands allow the user to define new positions in between two existing positions or at the beginning or end of a set of existing positions. These commands are also used for the initial definition of positions in a new file. The *Change* commands allow the user to change a subset of previously defined positions. The *Copy* command prompts the user for a subset of existing positions to be copied to a new location in the list of positions. Any existing positions in the list are moved down the list to make way for the new positions. The *Delete* command will erase a specified subset of positions. The *Display* command will display a specified subset of positions on the terminal screen, while the *Print* command will print the specified positions on the printer.

The last two commands in PODS' main menu deal with the creation and execution of a path. A path is a general shape or template which is useful in assembly operations (see Section 5.3.3). A set of positions may be made into a path by translating and rotating the set so that the robot box frame becomes the path coordinate frame (see Appendix A.1 for a definition of the path coordinate frame). After a path has been defined, it may be executed between two points defined in the workspace using the *Execute Path* option.

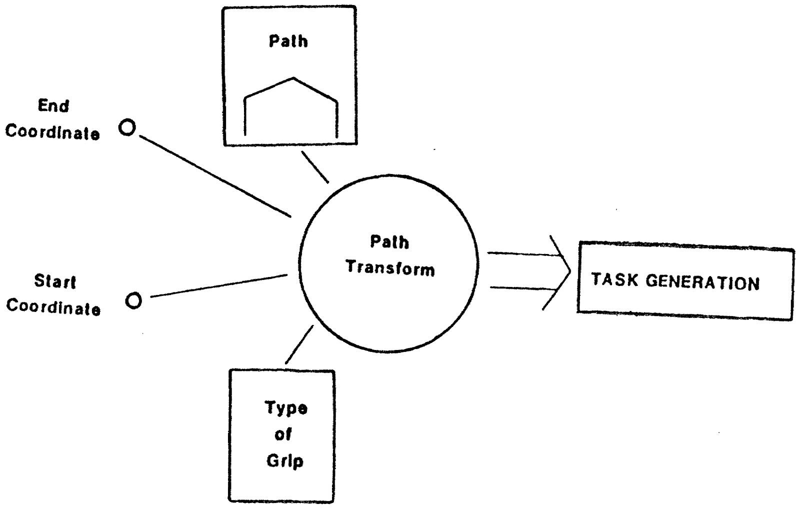


Figure 5.10: Task execution system.

5.3.2 Generation of Robot Paths in an Assembly Cell

A path definition system has been developed to simplify the teaching of robot assembly tasks. The system allows the robot controller to generate most of the details involved in the execution of an assembly task. Such a system allows much of the programming for a complete assembly task to be done off-line at a fairly high level. With this technique, the majority of robot action commands in an assembly task programmed by the user would be conveyed as tasks (such as getting or putting a part) instead of as point to point moves. By so doing, the controlling computer will automatically perform operations such as simple obstacle avoidance, part presence checking, gripper opening, speed control, and some error correction.

The basic elements used in generating tasks are shown in Figure 5.10. A path is a set of robot positions in the workspace which are used as a shape template for the task. The path transform performs a geometric and in some cases functional transformation of a path. A geometric transform translates, rotates, scales and sometimes skews the path such that the resulting curve is shaped similar to the original path, but connects two newly specified starting and ending positions. A functional transformation allows the path to be used as a template for an assembly task (i.e. getting a part from a fixtured position).

5.3.3 Path Definition

A path is defined as a set of two or more robot configurations in the workspace. Only the X, Y, and Z coordinates (this is the 7565 BOX frame [98]) are used as the template for subsequent transformations. Rotary axes are handled separately. The path is most easily defined using PODS (see Section 5.3.1). Paths are used as general shape templates for subsequent moves through trajectories in the workspace. Since only the shape of the path is used, the first location is arbitrary and is defined to be robot coordinates <0,0,0>. For many assembly applications it is useful to have a path library from which to select the appropriate path. Different path shapes may be selected for simple obstacle avoidance and minimal trajectory execution times. A more rigorous definition of a path may be found in Appendix A.1.

5.3.4 Path Transforms

Before a path becomes useful in an assembly environment, it must be operated upon by a path transform. The inputs to a path transform are shown in Figure 5.10. The path transform translates, rotates, scales and sometimes skews a path so that the first and last points of the path correspond to the start and end positions of the desired trajectory (see Figure 5.11). Since the two end points do not uniquely define a reference frame into which the path should be transformed, the path is constrained not to rotate along the line through the start and end points. With this constraint, the location of the two end points is sufficient to scale translate and rotate the path into the proper trajectory. For the cylindrical transformation, it is also necessary to specify an approach vector at each end point.

To date, four path transforms have been defined and found to be useful in an assembly environment. A fifth more general path transform, the cylindrical transform, has been developed but not implemented.

Spatial Path Transform

The spatial path transform rotates the selected path (X, Y, and Z coordinates only) into the reference frame defined by the start and end trajectory positions and equally scales the X, Y, and Z local coordinates of the path so as to make the first and last points of the path coincide with the start and end points of the desired

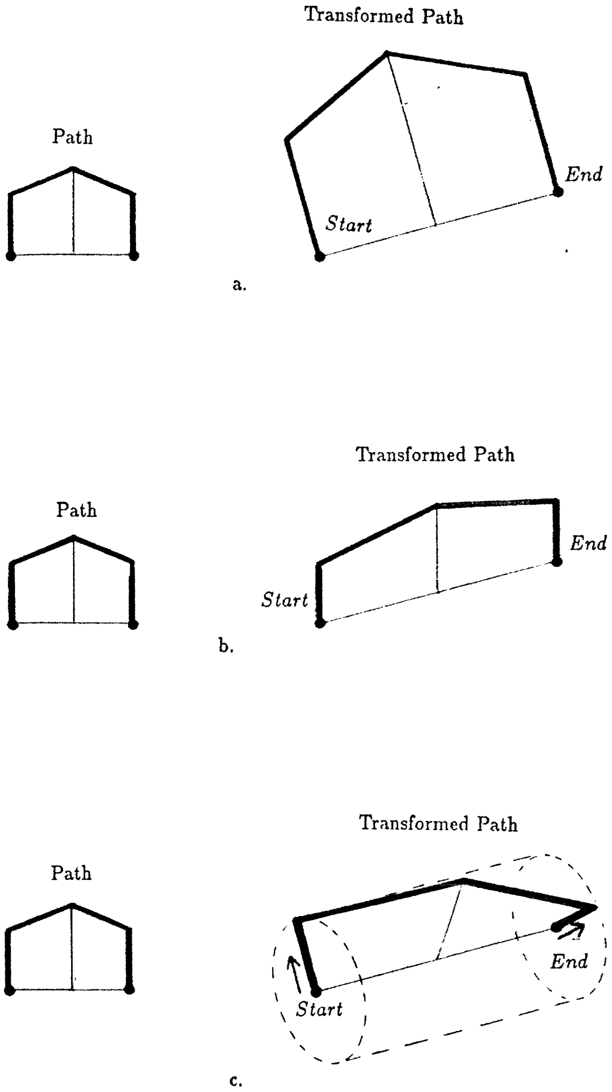


Figure 5.11: path transforms. a. Spatial path transform. b. Planar path transform. c. Cylindrical path transform.

trajectory (see Appendix A.2.1 for a more rigorous definition). An example of a spatially transformed path in two dimensions is shown in Figure 5.11a. In many assembly tasks, the robot wrist configuration is not dependent upon the X, Y, Z trajectory of the manipulator. If the wrist angles were rotated with the direction of the trajectory, parts fixed in the workspace would be gripped at different locations as a function of the trajectory. This is clearly undesirable. A fairly simple technique is used to address the requirement of retaining wrist angles with different X, Y, Z. trajectories. The wrist angles for the trajectory are taken from the desired start and end manually taught trajectory points and have no relation to the path or the spatial transformation of the path. The rotary joint angles from the start position are used for the first half of the trajectory positions and those from the end position are used for the second half of the trajectory.

Planar Path Transform

The planar path transform performs the same function as the spatial path transform except that the final trajectory shape is skewed from the original path shape such that all Z direction motions between path points remain Z direction motions in the trajectory. (See Figure 5.11b.) In addition, the local Z coordinates of the path (the local Z distance from a path point to the line through the starting and ending path points) remain unchanged after transforming. Thus, large excursions across the workspace do not cause increases in local Z motions. The wrist joints during a trajectory are configured as in the spatial path transform. The planar path transform is useful in assemblies with Z direction insertions. Since the Z coordinates of the path remain unchanged, the Z insertion phase of a task remains independent of the location of the end points. The planar path transform is a degenerate case of the cylindrical path transform with Z direction unit approach vectors.

Cylindrical Path Transform

The cylindrical path transform is very similar to the planar path transform but somewhat more general. Two approach vectors in addition to the two end points must be specified to cylindrically transform a path to a trajectory (see Figure 5.11c.). The starting approach vector specifies the departure direction from the starting trajectory point and the ending approach vector specifies the direction from which the trajectory is to approach the ending trajectory point. The second point of the path is positioned along the starting approach vector at a distance equal to the local Z distance of the path. The next to last trajectory point is similarly spaced from the ending trajectory point. The intermediate path points are helically positioned so as to join the second and next to last trajectory points (see Appendix A.2.2 for a more rigorous definition).

Get Path Transform

The get path transform is a functional as well as geometric path transform. It calls the planar path transform for the necessary geometric path transformation and adds the gripper motions necessary for grasping a part at the end of the trajectory. In addition to the two end trajectory points, the get path transform must be passed a flag which signifies an external or internal grip on the part. This path transform also limits the speed at which the robot grasps the part so that the dynamics of the part do not significantly affect the grasping process.

Put Path Transform

The put path transform is similar to the get path transform except the gripper is configured to grasp the part during the planar path transformed trajectory and release the part at the completion of the trajectory. The speed of withdrawal of the part (the section between the first and second trajectory points) is limited by the put path transform so that the robot dynamic forces are insignificant and don't jam the part during its removal from a pallet. During the grasping phase of the trajectory, the put path transform checks the strain gages in the grippers for the proper gripping force. If an unexpected force is found, an error recovery subroutine is automatically called by the put path transform (see Section 5.3.6).

5.3.5 Automatic Pallet Indexing

For assembly tasks where pallets are used to hold parts, the robot will need to access similar parts from different locations in a pallet. It is assumed that the parts are very accurately spaced in their pallets. The location of each pallet in the workspace is recorded by storing a single set of robot joint configurations which position the robot at a known reference point on the pallet. The location of the robot with its gripper positioned at the first part position of a pallet is stored in a pallet file. This configuration (the part grasping configuration) defines the location of the local pallet frame. The other parts in the pallet are accessed by adding an X, Y, Z offset to the origin of the frame. The information in a pallet file contains

- The name of the part being held
- The robot configuration at the first part position
- The X, Y, Z, offset between part locations
- The maximum number of parts in the pallet

In an assembly program, the position of the next available part in each pallet is stored in a globally defined aggregate. The pallet indexing subroutine uses the current pallet location for the goal of the next move and then indexes the current pallet location variable for that pallet. When the final pallet location is reached, the current pallet location is reset to the first location. In a real assembly environment an instruction would also be issued for the pallet changer to exchange the empty pallet for a full one.

5.3.6 Error Recovery

During the portion of an assembly task when the robot is in contact with its environment (i.e. holding parts), force sensing may be used to assure proper task completion. A number of error recovery algorithms have been designed which use the path definition system described in Section 5.3.2. Four different types of assembly procedure errors are handled by this system.

- Part missing at pallet
- Part slips during withdrawal from pallet
- Part dropped during trajectory from pallet to assembly
- Part dropped prematurely over assembly

If after an error correction procedure has been initiated another error occurs, the new error is handled before the first in a nested (recursive) fashion. The error recovery subroutine described in the present section is only called during a trajectory using the put path transform (a *put* task) which occurs only after a trajectory using a get path transform (a *get* task).

Part missing at pallet. If the error recovery subroutine is called before the first move of a put path transformed trajectory, it is assumed that no part is present in the current pallet location and a *get* task is issued to the next position of the current pallet. Upon the completion of the *get* task, the *put* task in which the error occurred is reissued.

Part slips from gripper during withdrawal from pallet. If the error recovery subroutine is called during a move from the first to second trajectory point during a *put* task, the subroutine infers that the part is stuck on the pallet. The subroutine reissues a *get* task to the same pallet location to try and regrab the part. If the part is not found or if it slips from the gripper for a second time during the reissued *put* task, a *Part missing at pallet* error recovery procedure is issued.

Part dropped during trajectory from pallet to assembly. This procedure is followed if the error recovery subroutine is called during a trajectory move between the second and second to last trajectory points. It is assumed that the part has dropped in an out of the way place and the *Part missing at pallet* procedure is followed.

Part dropped prematurely over assembly. If the error recovery subroutine is called during the final move of a put path transformed trajectory, it is assumed that the part has been dropped into or near the assembly. Since the computer cannot be sure where the part has been dropped and whether or not it might interfere with the operation of the final assembly, the error recovery subroutine halts execution and calls for human help before proceeding. The operator is instructed to either clear the part and restart the operation or to allow the robot to continue normally.

5.4 Assembly System Implementation

In a factory flexible-assembly environment, a number of the multifunctional assembly modules or cells could be used to perform the assembly or subassembly of one or more products. Prior to the start of a new assembly task, a cell would be put through a setup procedure designed for that particular assembly or subassembly. This setup phase may consist of the manipulator changing the end effectors on fixtures and jigs, moving fixtures around the cell, recalibration, and changing its fingers to accommodate the new parts. Following the setup phase, a new assembly procedure would be loaded into the cell controller for the next subassembly to be assembled. The assembly system described in Sections 5.2.1 and 5.3 are used to demonstrate such a factory assembly environment. In the setup phase the 7565 installs the proper fixturing in the assembly vise, positions the flexible fixtures around the workcell, and performs the calibration. The calibration procedure involves calibrating the rotary and linear joints of the robot to the workspace (see Section 5.4.2). The assembly procedure utilizes the hardware and the path definition software described previously to assemble and final test a number of consumer hand drills.

The software system for the cell was designed such that there is an integration between off-line and on-line task programming. The majority of the assembly procedure for the cell is programmed off-line using the AML subroutine modules described in Section 5.3. Selecting the robot grasp points and debugging the final program are performed on-line.

5.4.1 Assembly Cell Setup Procedure

The initial state of the assembly cell prior to the setup phase of the drill assembly consists of a disassembled assembly vise (plates and jaws removed) with the vise components fixtured in their storage magazines, flexible fixtures with end effectors retracted and the robot arm positioned far from obstacles.

A flowchart for the setup procedure of the prototype assembly cell for the assembly of a hand drill is shown in Figure 5.12. When the robot moves from one procedure to the next, the moves are buffered from one another so that the robot will not collide with its environment. The buffering operation involves the robot moving into a "freeway" area which is defined as the space above the assembly hardware (positive Z direction). The setup procedures in Figure 5.12 are defined as follows:

Check Gripper Status. After entering the freeway, the robot moves to an empty area of the finger magazine and checks to see which fingers are currently attached.

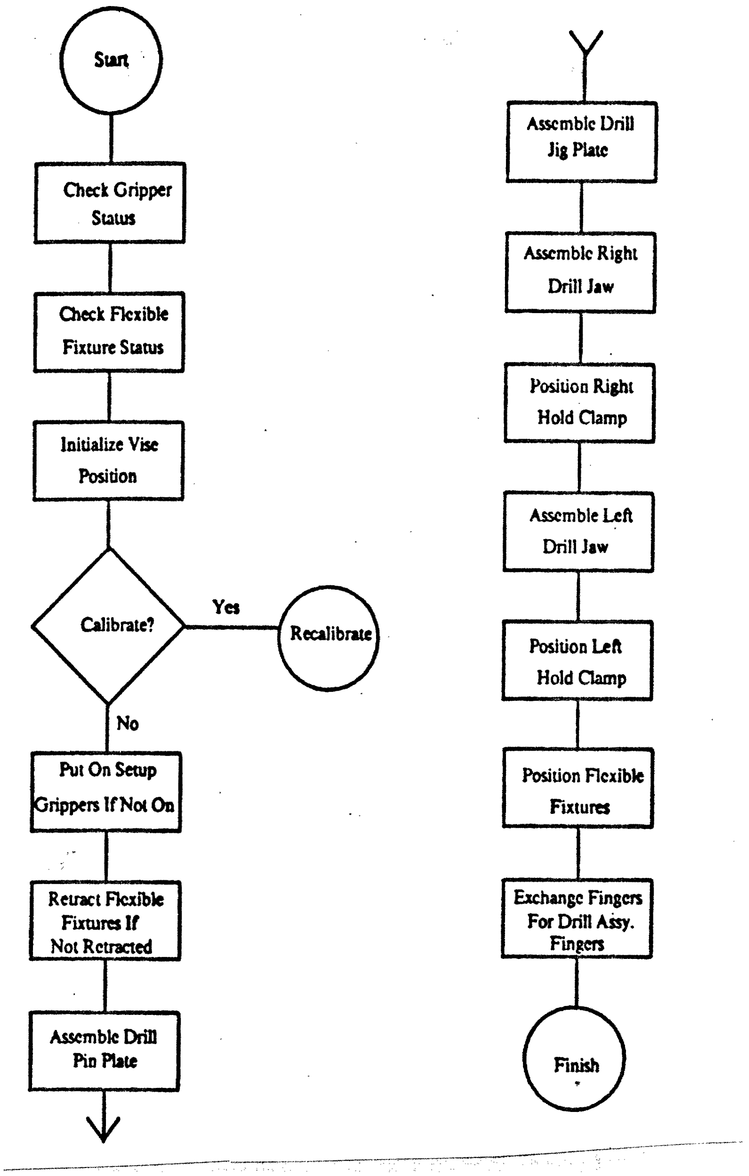


Figure 5.12: Assembly cell setup procedure.

Each set of fingers is a different length and the checking operation is performed by initiating a pinching motion at various heights above a bar. When the presence of the bar is sensed (fingers make contact), the robot can determine which fingers are mounted by checking its Z coordinate.

Check Flexible Fixture Status. The Series-1 computer prints an inquiry message to the user as to the initial location of the flexible fixtures (they are either positioned, retracted, or lost). The user is instructed to type in the proper response to the inquiry. If the fixture is lost, a flexible fixture positioning subroutine must be called. This subroutine sets the proper fixture to the slack condition and allows the user to insert the flexible fixture gripper interface into the setup fingers of the robot. The robot then positions and locks the fixture in the proper location.

Initialize Vise Positions. In this procedure, the computer instructs the user to make sure that all components are removed from the assembly vise and properly positioned in their respective magazines. The computer then actuates the vise so that it is in the proper configuration for the rest of the setup procedure.

Calibrate. The user is asked if the system needs calibration (see Section 5.4.2). If not, the next setup procedure is performed. If the system needs to be calibrated, the computer will first determine if the fixtures need moving in order for the robot to access the calibration posts. If the fixtures need moving the robot will first don the setup fingers then proceed to move the fixtures to their retracted positions. The calibration procedure described in Section 5.4.2 is then performed.

Put On Setup Gripper Fingers. If the setup fingers are not on the robot grippers, the robot will take off any fingers it is wearing and put on the setup grippers using the technique described in Section 5.2.1.

Retract Flexible Fixtures. If the flexible fixture which is near the assembly vise is not in its retracted position, the robot will perform a fixture approach, positioning and deproach subroutine to place the fixture into its retracted configuration.

The next few operations involve the robot grasping an assembly vise component and inserting it into the vise.

Assemble Pin Plate. The pin plate is grasped in the center portion of the grippers where there is a section of elastomeric material. The robot is then moved through a set of previously taught points which extract the pin plate from its fixture, move it to the assembly vise and assemble it into its proper location. The pin plate, as well as all of the other plates and jaws for the vise, are designed to be extremely easy for the robot to assemble (large bevels and chamfers are on mating features). Because of this, the assembly of these components has proven to be extremely reliable and no force sensing is necessary to verify task completion.

Assemble Jig Plate. The jig plate assembly proceeds much like the pin plate assembly except that the jig plate itself has elastomeric material mounted on its gripper mating surface to enhance grasp stability.

Assemble Drill Jaws. Each of the jaws used in the assembly vise to clamp the base part of the drill assembly has two accurately positioned pins which plug into the setup gripper fingers. Thus, each jaw is accurately located in the robot's grippers prior to insertion into the assembly vise. Each jaw has an accurately machined dovetail on its rear surface which slides into a mating dovetail on the vise carriage. The dovetail on the vise carriage has a chamfered-dovetailed lead-in section as well as a cam actuated gib to lock the jaw in place. After the robot places the jaw in position it actuates a lever which pushes the gib against the jaw's dovetail to accurately lock it in place.

Position Hold Clamps. The robot uses the tips of its setup fingers to rotate the proper hold-down clamp into position.

Position Flexible Fixtures. Here the controller runs a program which reverses the actions used in the *Retract Flexible Fixtures* procedure.

Exchange Fingers. The robot executes its finger changing moves to replace the drill assembly fingers for the setup fingers.

5.4.2 Workspace Calibration Procedure

While performing experiments on close tolerance assembly operations, a time dependency of robot repeatability was found. It was proposed that a calibration of the robot joints would improve the performance of close tolerance tasks. If the workspace happened to be displaced with respect to the robot BOX frame, it would still be possible to perform tasks after a recalibration. In the workspace calibration procedure, all six axes of the robot are calibrated. The calibration procedures for the yaw and pitch axes use two straight edges fixed in the workspace. The its Y axis. The fixture was moved until it aligned with the axis within .002 inches. The pitch axis is calibrated at a 60 degree angle with respect to the reference base plates. An accurate 60 degree triangle was used as a straight edge at the proper angle. The roll axis is calibrated with a single post in the workspace using the CALROLL subroutine. Before the CALROLL procedure is performed, the yaw and pitch axes are set to their calibrated values.

Linear Joint Calibration

The location of the workspace in robot X, Y, Z coordinates is found through the location of 3 posts in the workspace. Any changes between the robot linear axes and the workspace location are assumed to be rigid body displacements and rotations. The robot locates the workspace by using the AML FINDPOST subroutine [98] to locate 3 posts numbered 0 through 2. The vector from post 0 to 1 defines the Y workspace axis and the vector from post 0 to 2 defines the X workspace axis. Before the linear axes are calibrated, the yaw and pitch axes are set to their calibrated values.

Assembly Environment Calibration Procedure

Each time a set of positions are stored, a set of post locations are associated with it (see Section 5.3.1). In an assembly procedure, the transformation in BOX frame coordinates from the location of 3 stored posts to newly measured locations of the same 3 posts is used to calculate a correcting transformation when playing back the points. Another technique of storing the transformation data was considered. This second technique involves finding the transformation from the workspace frame as defined by the 3 posts to the BOX frame and then performing an inverse transformation of the location of the robot joints prior to storing them. When playing them back, the forward transformation from the newly found workspace frame to the BOX frame is used. Because the second technique transforms the positions so they no longer appear as robot joint coordinates in the file, it becomes more difficult to edit the points and was therefore not used.

The rotary joint calibrations are used in a slightly different fashion than the X, Y, Z calibrations when making joint position corrections. The values returned from the rotary calibration procedures are the rotation angles which will position the joints to the measured 0 location relative to the calibration fixtures. In general, all values returned are relatively small. The calibration angles measured at the time the positions are stored are subtracted from the positions before they are recorded in a file. When the positions are played back, a new set of rotary offsets are added to the roll, pitch and yaw joint angles before executing a robot action.

Variation of Calibration Measurements

7565 repeatability tests were performed to determine how robot repeatability varied over time. At the time the test was performed, only the linear calibration fixtures were in place, so no rotary calibration was done prior to measuring the location of the posts in the workspace. The test consisted of exercising the robot and then instructing it to perform a FINDPOST procedure on three posts. The entire cycle lasted about 6 minutes and was repeated for 2 hours. The test results showing the measured X, Y, and Z locations of the 3 posts relative to their nominal positions measured during 2 hours of operation for two separate runs is shown in Figure 5.14a through 5.14c. Each run was started with the robot hydraulics at room temperature and the computer and I/O electronics warmed up for at least 24 hours. The smallest variations were in the Y axis direction – about .006 inches maximum. The largest excursions in positions were seen in the Z axis direction which varied .016 to .020 inches in the first 45 minutes of the test and then varied less than .003 inches for the duration of the test.

The results of the test show that there appears to be a warm up period before the robot can be considered repeatable to less than .015 inches. A warm up time of about 45 minutes conforms to the suggested time in the AML manuals [98]. The source of the time dependent errors has not been investigated to date, but is likely to be temperature sensitivity of the tempo-sonic linear position transducers, potentiometers, the servo electronics and mechanical structures or some combination of these elements. A temperature sensitivity of the tempo-sonic transducers may explain the large variations seen in the Z axis and the smaller variations in the X and Y axes because the Z axis transducer is the only one positioned next to incoming hot hydraulic oil. The larger variation in the X axis compared to the Y axis may be due to the temperature sensitivity of the pitch potentiometer since the pitch axis is aligned with the Y axis during the entire test calibration procedure.

After numerous calibrations were performed prior to high tolerance assembly

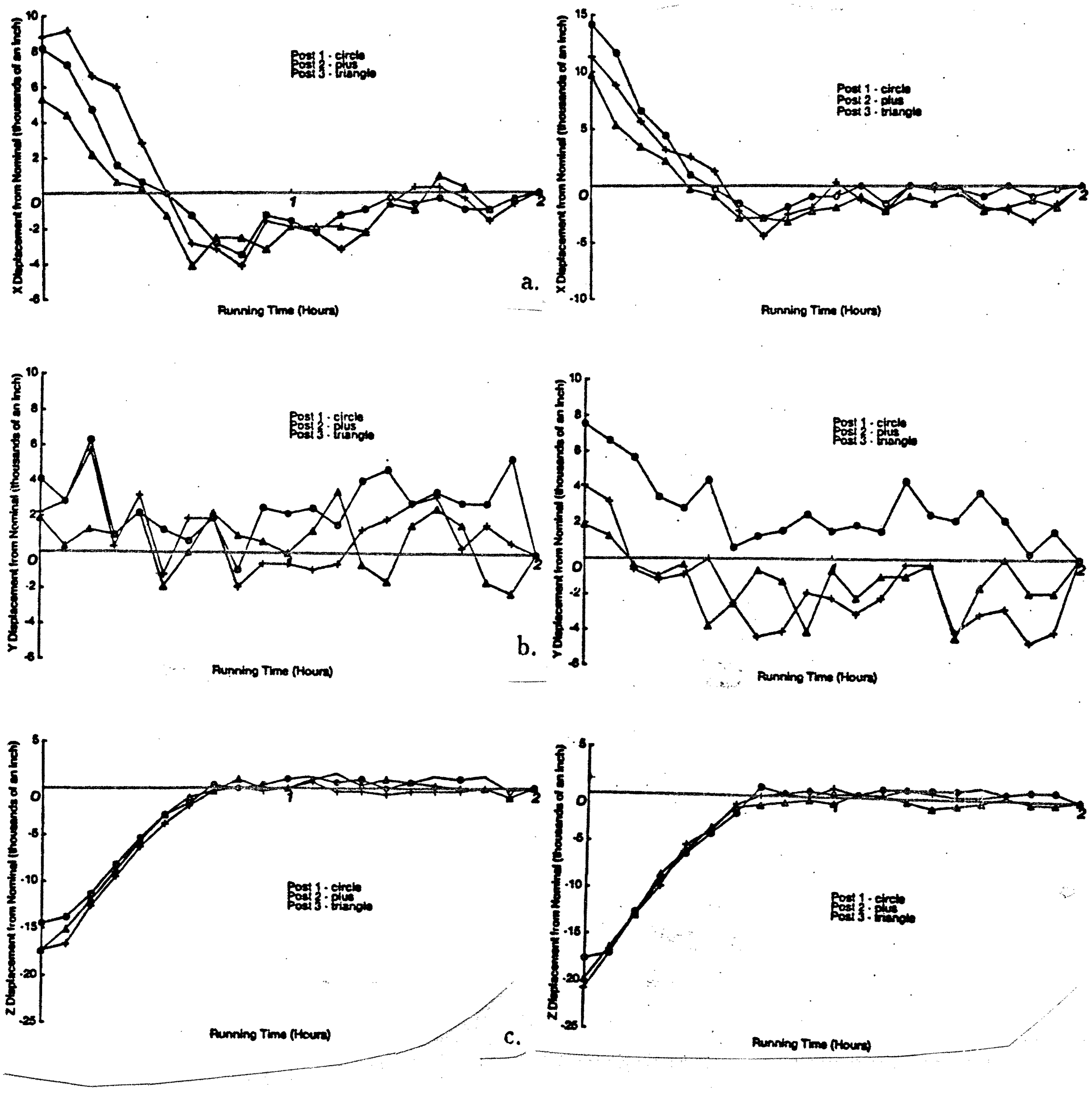


Figure 5.14: Robot repeatability vs. time for each manipulator direction. a. x axis, b. y axis, c. z axis.

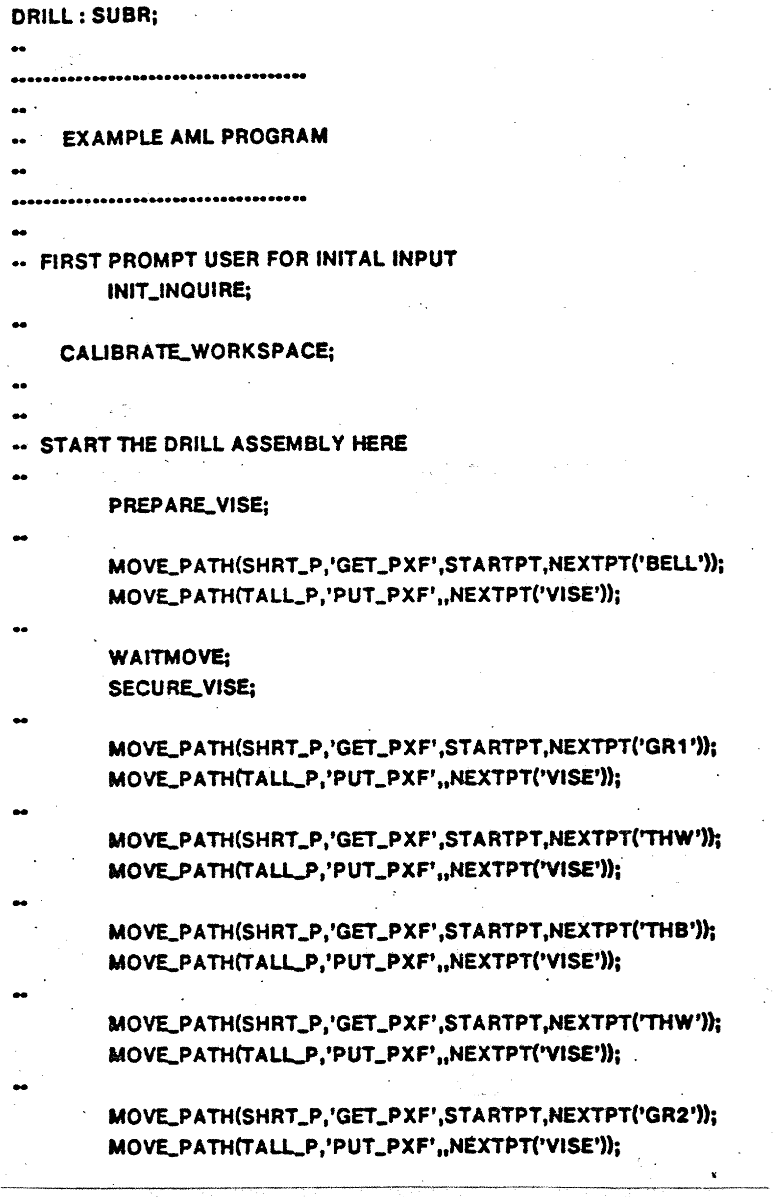
tasks, it became apparent that the robot was incapable of locating its workspace to a sufficient accuracy for the successful completion of the tasks. It was also noted that when the robot was warmed up for more than about 45 minutes, many of the high tolerance assembly tasks could be performed as long as no calibration procedure preceded the task. Thus, the robot could not locate its workspace to the same accuracy as its steady state (warmed up) repeatability. At times it was as much as .030 inches (.75 mm) out of calibration. Position errors often appeared to be larger than the calibration offsets as the robot changed its joint configurations for different phases of the assembly tasks.

5.4.3 Power Drill Assembly Procedure

The assembly of Black and Decker $\frac{3}{8}$ inch hand drills was used to demonstrate some of the assembly hardware and software concepts in Sections 5.2.1 and 5.3. The drill assembly task consists of the mechanical insertion of 13 of the 19 parts of the drill, Figure 5.15. Electrical wiring, switch, brush and handle assembly operations are not addressed. The layout of the drill assembly cell is shown in Figure 5.16. The assembly procedure is coded in AML and uses the path definition system described in Section 5.3.2. The majority of the assembly procedure is programmed off-line. Only the gripper grasp point for the parts need be programmed on-line.

The top level cell control program is generated off-line by a programmer. An example of part of a drill assembly program is shown in Figure 5.17. Each MOVE_PATH instruction is essentially a task level instruction for the robot. The arguments to this instruction contain information about the type of task to be performed, the shape of the path through which the robot should move, the type of grip to be used on the part and the part with which the task should be performed. MOVE_PATH instructions with alternating calls to GET_PXF and PUT_PXF (get and put path transforms) comprise the skeleton of the assembly program. Additional commands for calling subroutines which control peripheral hardware (devices other than the robot) make up the majority of the remainder of the assembly procedure.

The assembly procedure is shown in flowchart form in Figure 5.18. Most of the operations involve the acquisition and the insertion of the appropriate part in the assembly. Robot motions corresponding to the *get* then *put* instructions in Figure 5.18 consist of the robot moving through the desired path to the proper



```
DRILL : SUBR;
--
--------------------------------------------
--
--    EXAMPLE AML PROGRAM
--
--------------------------------------------
--
-- FIRST PROMPT USER FOR INITAL INPUT
          INIT_INQUIRE;
--
     CALIBRATE_WORKSPACE;
--
--
-- START THE DRILL ASSEMBLY HERE
--
          PREPARE_VISE;
--
          MOVE_PATH(SHRT_P,'GET_PXF',STARTPT,NEXTPT('BELL'));
          MOVE_PATH(TALL_P,'PUT_PXF',,NEXTPT('VISE'));
--
          WAITMOVE;
          SECURE_VISE;
--
          MOVE_PATH(SHRT_P,'GET_PXF',STARTPT,NEXTPT('GR1'));
          MOVE_PATH(TALL_P,'PUT_PXF',,NEXTPT('VISE'));
--
          MOVE_PATH(SHRT_P,'GET_PXF',STARTPT,NEXTPT('THW'));
          MOVE_PATH(TALL_P,'PUT_PXF',,NEXTPT('VISE'));
--
          MOVE_PATH(SHRT_P,'GET_PXF',STARTPT,NEXTPT('THB'));
          MOVE_PATH(TALL_P,'PUT_PXF',,NEXTPT('VISE'));
--
          MOVE_PATH(SHRT_P,'GET_PXF',STARTPT,NEXTPT('THW'));
          MOVE_PATH(TALL_P,'PUT_PXF',,NEXTPT('VISE'));
--
          MOVE_PATH(SHRT_P,'GET_PXF',STARTPT,NEXTPT('GR2'));
          MOVE_PATH(TALL_P,'PUT_PXF',,NEXTPT('VISE'));
```

Figure 5.17: Example top level assembly program.

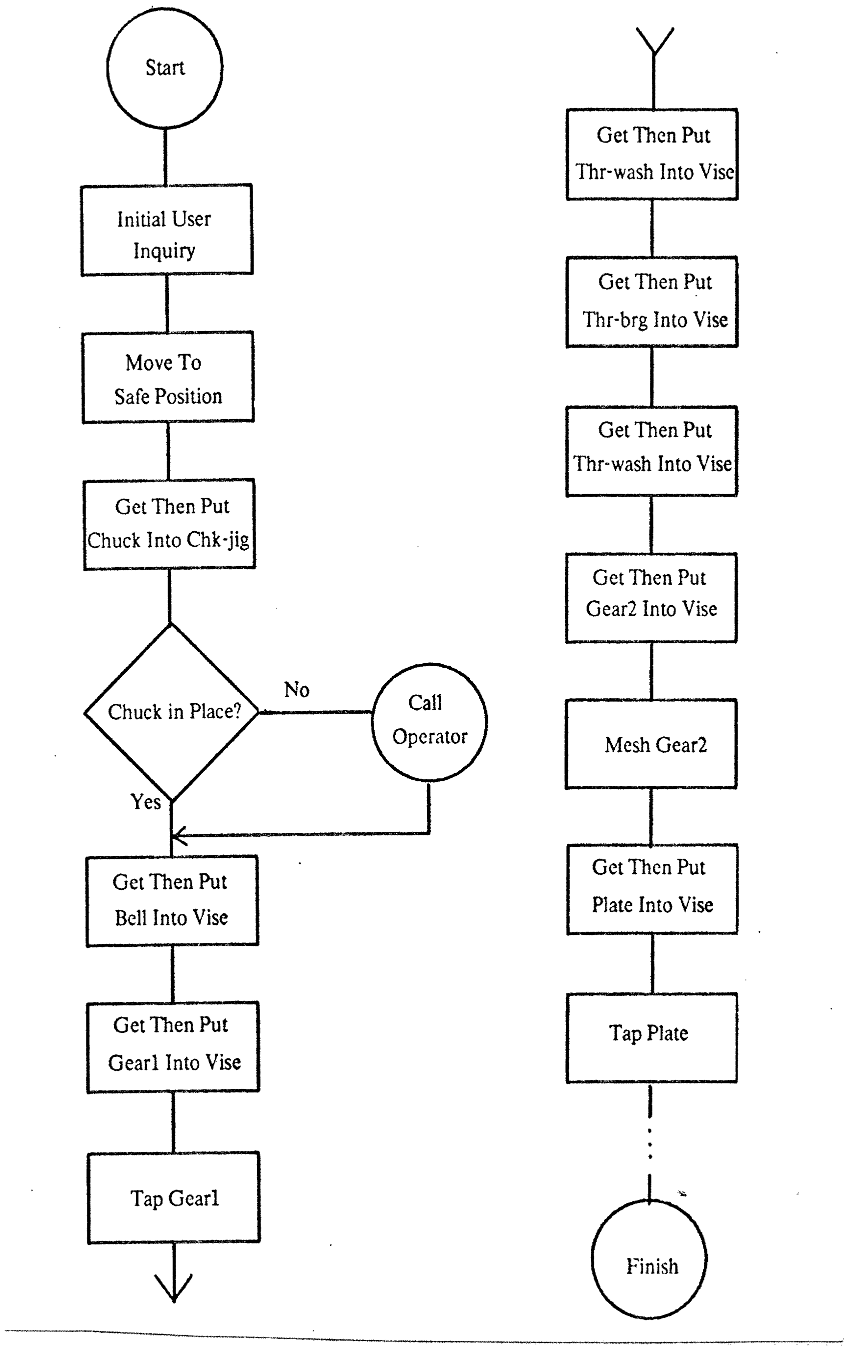

Start
Initial User Inquiry
Move To Safe Position
Get Then Put Chuck Into Chk-jig
Chuck in Place?
No
Call Operator
Yes
Get Then Put Bell Into Vise
Get Then Put Gearl Into Vise
Tap Gearl
Get Then Put Thr-wash Into Vise
Get Then Put Thr-brg Into Vise
Get Then Put Thr-wash Into Vise
Get Then Put Gear2 Into Vise
Mesh Gear2
Get Then Put Plate Into Vise
Tap Plate
Finish

Figure 5.18: Drill assembly procedure.

Drill Assembly Operations		
Part Name	Assembly Fixture	Type of Tap Required
Bell Housing	Vise	
Gear 1	Vise	Seating
Thrust Washer	Vise	
Thrust Bearing	Vise	
Thrust Washer	Vise	
Gear 2	Vise	Meshing
Bushing Plate	Vise	
Soft Washer	Flexible Fixture	
Rotor	Vise	Meshing
Soft Washer	Vise	Low Tolerance Mating
Shroud	Stator Pallet	Seating
Stator	Vise	
Chuck	Chuck Jig	

Table 5.1: Hand drill parts in order of assembly.

the chuck to the bottom of the drill (see Figure 5.19). In this operation, the flexible fixture is used to locate and actuate the end effector which applies power to the drill. This device has a standard flexible fixture end effector interface. It also has contacts which plug into the stator windings and brushes which are actuated by solenoids to contact the commutator of the rotor. A micro-switch sensor is mounted under the chuck in the chuck jig and senses when the chuck has been fully threaded onto the shaft. This sensor also acts as a final test to check the proper operation of the drill motor.

The intermediate part fixturing function of the flexible fixtures is demonstrated in the assembly of the first soft washer. Just after the acquisition of the soft washer, the flexible fixture is extended to a position over the assembly to provide an intermediate fixturing location. The washer is then inserted into the flexible fixture. Next, the rotor is acquired and inserted through the fixtured washer. The flexible fixture is then retracted and the rotor is positioned to its proper position in the assembly. This procedure was necessary for the reliable assembly of the soft washer. If the washer were directly assembled onto the previous part (the bushing plate), it would be unstable (see Section 2.2.1 for a definition of an unstable assembly).

Aspects of the assembly procedure which are taught on-line include gripper two trajectory points is not appropriate for obstacle avoidance, another path should be selected or defined on-line. It was found that most of the on-line programming effort was in teaching reliable grasping geometries between the robot and the parts and modifying the pallets to accurately fixture the parts without over constraint which may cause jamming upon part removal. The basic assembly procedure (top level program) remained relatively unchanged during the debugging phase with the exception of the addition of some "tap" operations for a few parts.

5.5 Conclusions and Discussion

After studying this and past prototype programmable assembly cells, a number of conclusions have been drawn and a list of issues which have yet to be fully addressed has been formulated.

1. Only a small subset of assembly tasks are possible with most single armed robots. Special assembly tools and end effectors are required for many tasks.

2. High precision assembly (clearances between mating parts from 0.25 to 1.0 mm) is possible at moderate to fast speeds (1 meter per second) using position feedback only and the following stipulations

   (a) Joint stiffness must be high enough for fast mechanical settling, but low enough to allow adjustments for part misalignments with negligible interpart forces. The location of the rotational compliances strongly affects which operations may be performed as well as the speed at which the operations may be performed.

   (b) Adequate manipulator repeatability. This quantity depends upon the size of the part chamfers, the manipulator's compliance, and the stability of the manipulator's grasp on the part. For parts which are held relatively firmly (not easily displaced from the gripper) and have relatively large chamfers (about 1 mm), assembly with clearances as low as 0.01 inches (0.25 mm) is possible with the 7565 (robot repeatability $\approx$ 0.1 mm). Smaller clearance insertions such as the .002 inch (.05 mm) clearance first-gear shaft insertion was possible if the parts were allowed to fall together then tapped into place. The compliance of the 7565 is difficult to accurately measure due to the robot's integral position controller.

(c) Certain operations which involve the interaction of the manipulator or part being held by the manipulator with a "stiff" environment must be performed at reduced speeds for reliable operation. Also because the grippers were often tipped 45 degrees form their rest position, Z direction accelerations produced appreciable X and Y direction dynamics induced deflections and had to be limited.

(d) Part grasping and extraction is difficult for accurately fixtured parts and must be performed so that the point of application of the withdrawing force is in a location which minimizes the possibility of wedging or jamming [201]. For a part with a cylindrical hole fixtured on a close tolerance pin, the center of compliance [104,133] of the manipulator-part system should be just above the top of the fixturing pin. If such a grasp location cannot be found, an offset moment may be applied to the part to facilitate its extraction.

3. Force feedback controlled assembly and force monitored assembly (guarded moves [113]) cannot be performed with many commercial robots at speeds which are appropriate for high speed industrial assembly. A very high bandwidth system is required (see reference [202] for an overview of force control).

4. The cost of performing operations inside a robotic assembly cell is extremely expensive; therefore, time inside the cell needed to perform operations other than assembly (such as gripper changing) should be minimized.

5. A significant amount of time in an assembly task is spent waiting for the manipulator or other mechanical devices to come to rest.

6. Development time needed to automate an assembly may be reduced by performing most of the assembly cell programming tasks off-line.

7. Tasks such as teaching adequate manipulator grasp points for parts is extremely difficult to perform off-line due to unforeseen gripper deflections and part-fixture jams during part removal.

8. Robot fixture tooling may be readily changed by a robot such as the 7565, but should be specially designed for ease of assembly (and disassembly).

Issues which were cited as requiring much additional research include

1. The use of sensors to eliminate much of the required accuracy in fixtures, pallets, grippers and other hardware upstream of the assembly cell.

2. Investigation of the effect of manipulator impedance on the speed and ease of performing assembly operations.

3. Systems that will allow more complete off-line programming of an assembly task including geometric and other physical models.

4. Modeling of dynamic interaction of parts, manipulator, and environment and efficient modes of control for these interactions.

5. Systems which are capable of sufficiently accurate and reliable programmable part feeding.

6. The development of assembly tools to accomplish the operations not yet addressed.

Prototype Position-Sensor-Driven Assembly System

Chapter 6

A prototype part position sensor was designed based on analysis in Chapters 3 and 4. It measures the location of certain part features in six degrees of freedom based on triangulation of points illuminated by a light plane source.

The prototype was constructed and then calibrated on a test bed. The calibration procedure entailed quantifying relevant parameters for both the camera and the light source. Accuracy in measuring the location of a right corner feature was then determined through a battery of tests. Finally, the sensor was removed from the test bed and used with a mechanical manipulator to perform assembly tasks with real parts.

6.1 Components of the Prototype Sensor

The sensor consists of a helium-neon laser (Mells Griot model 05-LHR-151) and cylindrical lens (Mells Griot model LM-60), an MOS-type camera with 320 × 244 pixels (Hitachi model KP-120) and a 50mm lens (see Figure 6.1), a Symbolics 3600 Lisp machine and bit-graph terminal, a 454 × 576 × 8 bit frame-grabber (built at the MIT AI lab by Noble Larson), a hardware convolver (also built at the MIT AI lab by Noble Larson [134]) and a color graphics display. Both the camera and the frame-grabber are synced by the same clock. The camera constantly sends a video signal to the frame-grabber. When so instructed by the Lisp Machine, the frame-grabber fills its memory with the output from the camera. The pixel clock internal to the camera is not synced with the frame-grabber pixel clock; thus, the pixels in

The disparity angles (θ_{LP}) of the two lasers (only one laser is shown in Figure 6.2) were 64 and 68.5 degrees.

6.3 Calibration of the Camera-Light Stripe System

Both the camera optical and laser geometric parameters were calibrated on the test bed. The camera magnification factor was independently calculated along the two directions of the image plane. The apparent difference in the magnification factor was due to non-synchronized sampling along the horizontal direction (see Section 4.3). Magnification is defined as the length from the rear nodal plane to the image plane divided by the width of a pixel (see Section 4.2.1 for lens parameter definitions). The following camera parameters were calibrated

r_{pixel}: Apparent aspect ratio of the pixels in the frame-grabber.

x_{target}: Distance from the front nodal plane to the base position of the target.

(y_{COE}, z_{COE}): Sensor frame coordinates of the center of expansion.

$\frac{f_0}{L_{pix.y}}$: Magnification along the sensor y coordinate.

$\frac{f_0}{L_{pix.z}}$: Magnification along the sensor z coordinate.

It was necessary to calculate the base distance from the camera to the target, x_{target}, for accurate modeling of perspective projections.

The calibrated laser parameters (described in detail in Section 6.3.2) were (refer to Figure 4.2)

θ_{LP}: Disparity angle of the laser plane from the optical axis.

ϕ_{LP}: Tilt angle of the laser plane from the vertical.

y_{LP}: Distance from the center-of-expansion in the image plane to the laser plane along the camera y axis.

6.3.1 Camera Calibration Procedure

A single step procedure was designed for the camera calibration. A series of images of an optical target were recorded at different distances from the camera (along the approximate optical axis). A second set of images were taken at the same distances, but with the target rotated 90 degrees.

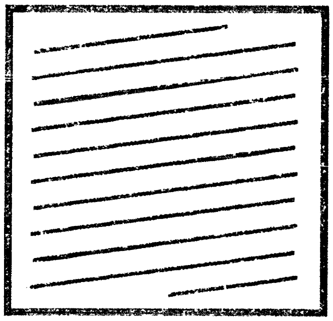


Figure 6.3: Sketch of target used for camera calibration. Lines are accurately spaced $\frac{1}{4}$ inch (6.35 mm) apart.

Camera Calibration Target

The target used to calibrate the camera parameters consists of a flat surface with a number of thin parallel lines drawn at a .25 inch (6.35 mm) spacings, Figure 6.3. The lines are purposely positioned so that they are not oriented along the axes of the pixel array of the camera. This avoids measurements at singularity positions identified in Section 4.3.

Pixel Aspect Ratio

The apparent aspect ratio of the pixels in the frame-grabber may be different from unity if the camera is not synced on the pixel level with the frame grabber. Most video cameras do not have an external pixel clock sync input, thus a slight distortion in image shape occurs after digitization in the frame grabber. Because of the asynchronous clocking of the frame grabber and the camera, this distortion is sensitive to environmental (temperature) changes.

The frame grabber aspect ratio may be calculated from two images of the target (call them *image 1* and *image 2*). Both images are taken with the target at the same distance from the camera, but one target image is rotated 90 from the other.

The distance between the lines in the i^{th} image, $D_{I,i}$, is a function of the distance between the lines on the target, D_{target},

$$D_{I,i} = \sqrt{(D_{pix,i} L_{pix,y} \sin\theta_i)^2 + (D_{pix,i} L_{pix,z} \cos\theta_i)^2} = \frac{D_{target} f_0}{x_i} \tag{6.1}$$

where θ_i is the angle of the target line, $D_{pix,i}$ is the distance between target lines in the image measured in pixels and x_i is the distance from the front nodal plane to the target. Taking the ratio $\frac{D_{I,1}}{D_{I,2}}$ equal to unity, and after some rearrangement, we obtain an expression for the pixel aspect ratio

$$r_{pixel} = \frac{L_{pix,y}}{L_{pix,z}} = \sqrt{\frac{\mathcal{S}_1^2 D_{pix,1}^2 (1 + \mathcal{S}_2^2) - \mathcal{S}_2^2 D_{pix,2}^2 (1 + \mathcal{S}_1^2)}{D_{pix,2}^2 (1 + \mathcal{S}_1^2) - D_{pix,1}^2 (1 + \mathcal{S}_2^2)}} \tag{6.2}$$

where we have used the relationships

$$\begin{aligned} \tan\theta_i &= \mathcal{S}_i \\ \sin\theta_i &= \frac{\mathcal{S}_i}{\sqrt{1 + \mathcal{S}_i^2}} \\ \cos\theta_i &= \frac{1}{\sqrt{1 + \mathcal{S}_i^2}}. \end{aligned} \tag{6.3}$$

Center-of-Expansion Calibration

The center-of-expansion is a point in the image plane of a camera which is defined by a certain straight line motion of a target (see Figure 6.4):

> Given an object plane (more specifically, a set of points which lie in a plane and are visible to a camera) which is not parallel to the optical axis of a camera and a motion of the object plane which is not parallel with the plane, the center-of-expansion is the point in the image plane which corresponds to a single point in the object plane throughout the motion.

As the target in Figure 6.3 moves away from the camera (motion should be roughly parallel to the optical axis), the lines in the image appear to separate from one another. An imaginary line parallel to the target lines which passes through the center-of-expansion will be the only line which appears not to move (this line will most likely be located in between a pair of actual lines on the target). This

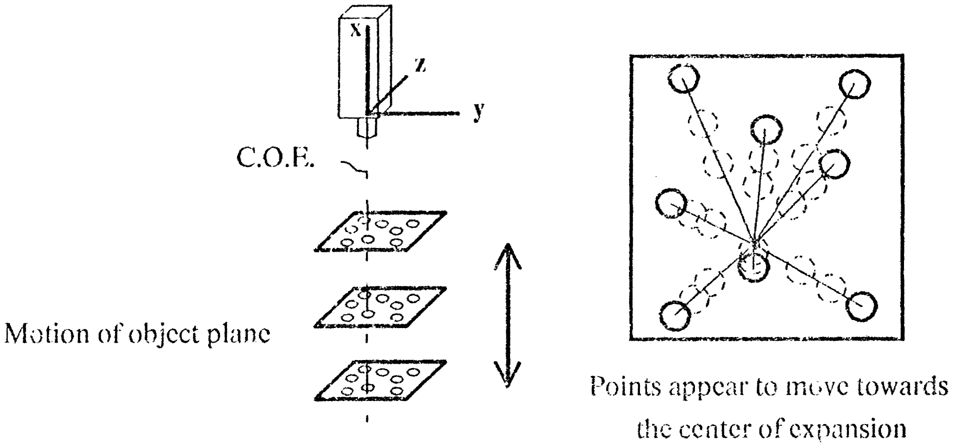


Figure 6.4: Camera and object plane used to define the center-of-expansion. Images are stored as the object plane is moved along the camera's x axis.

line is denoted as the stationary line. The intersection of stationary lines from two image sequences (the target for each sequence is rotated 90 degrees) defines the center-of-expansion.

The parameters of the stationary lines are calculated from pairs of images within each sequence. Results from the calculations from the image pairings are combined to obtain a sequence result. Two stationary line calculation sequence results are then used for the center-of-expansion estimate. The contribution from each line in the image is appropriately weighted (line parameter weights are $\frac{1}{\sigma^2}$ where σ^2 is the parameter variance) based upon its length. Equations (4.22) and (4.23) in Section 4.3 gives the relationships between segment length and parameter variances.

The ratio of perpendicular distances of a pair of lines in the image to the center-of-expansion is the same as the ratio of distances between lines in the corresponding images.

$$\frac{L_{1,actual}}{L_{2,actual}} = \frac{D_1}{D_2} \tag{6.4}$$

where $L_{1,actual}$ is the distance from the center-of-expansion to a certain line in the first image and $L_{2,actual}$ is the distance from the center-of-expansion to the image of the same line in the second image (the target has been moved to a different depth for the second image), and D_i is the weighted average distance between adjacent lines in the i^{th} image of the target. Measurements of distances to the lines are initially made from a point which is not the center-of-expansion since its location is not known *a priori*. Thus, the distance to the center of expansion from a certain line is

$$L_{i,actual} = L_{i,meas.} + \Delta L, \tag{6.5}$$

where $L_{i,meas.}$ is the perpendicular distance from the initial reference point to the line and ΔL is the perpendicular distance from the stationary line to the initial reference point. Thus, once ΔL is known, the location of the stationary line is known. Substituting Equation (6.5) into Equation (6.4) and solving for ΔL

$$\Delta L = \frac{D_1 L_2 - D_2 L_1}{D_2 - D_1}. \tag{6.6}$$

The slope of the stationary line in a sequence of images is the weighted mean slope of all of the lines in all of the images and the intercept is the weighted average intercept calculated from the intercept of each line and Equation (6.6).

An iterative procedure is used to find the center-of-expansion. An initial guess is made and the calculation described above is performed. Using a good first guess, the procedure converges after one iteration.

Calculation of the Base Position of the Target

Accurate direct measurement of the distance from the camera lens front nodal plane to the target is difficult to make since physical access to the lens is limited. The following is a technique for making this measurement from a set of images of a target of precisely spaced lines. The target (Figure 6.3) is positioned at precisely incremented distances from the camera. Because the incremental target positioning and line spacing is precise, an accurate range estimate may be obtained without knowledge of the lens parameters.

An arbitrary position of the target from the front nodal plane of the lens, x_{target}, is correlated to x settings on the test bed in order to obtain absolute distance measurements of the target. The distance between lines in the image plane (measured in pixels) for the i^{th} image as a function of the distance from the lens is given in Equation (6.1). We will consider calculations made from two images, say j and k, which correspond to test bed x position readings x_j and x_k respectively. A certain distance from the lens to the target, x_{target}, may be selected such that the actual distance from the lens to the target at positions x_j and x_k is $x_{target} + x_j$

and $x_{target} + x_k$ respectively. Taking the ratio of the distance between lines in two images (from Equation (6.1) and the appropriate choice of x_{target})

$$\frac{D_{pix,k}}{D_{pix,j}} = \frac{x_{target} + x_j}{x_{target} + x_k} \tag{6.7}$$

Solving for x_{target}

$$x_{target} = \frac{x_j - \frac{D_{pix,k}}{D_{pix,j}} x_k}{\frac{D_{pix,k}}{D_{pix,j}} x_k - 1}. \tag{6.8}$$

Pairs of images are used to generate estimates for x_{target}. These estimates are then combined using appropriate weightings. The weightings are based on the fitted line parameter errors for lines in the image of the target. Errors in manual positioning of the test bed were fairly small and an order of magnitude calculation showed they had little effect on the total error.

Magnification Calibration

The camera magnification calculation uses the same set of images as the center-of-expansion calculation. A separate magnification calculation was performed for each image and the weighted mean was used as the final value. The quantity used for the magnification includes the conversion from inches (mm's) to pixels. The apparent magnifications in the y and z directions are different if the pixel aspect ratio is not unity.

Using Equations (6.1) and (6.4) we may solve for the magnifications in the y and z image plane directions

$$\left(\frac{f_0}{L_{pix,y}}\right)_i = \frac{-x_i D_{pix,i}}{D_{target}} \sqrt{\frac{\frac{1}{r_{pixel}^2} + S_i^2}{1 + S_i^2}} \tag{6.9}$$

$$\left(\frac{f_0}{L_{pix,z}}\right)_i = \frac{-x_i D_{pix,i}}{D_{target}} \sqrt{\frac{1 + (r_{pixel} S_i)^2}{1 + S_i^2}}. \tag{6.10}$$

For good resolution, Equation (6.9) should be used on images of lines with large slopes ($S_i \gg 1$) and Equation (6.10) should be used on images of lines with small slopes ($S_i \ll 1$).

6.3.2 Calibration of Laser Parameters

The light plane calibration procedures are similar to those used for the camera calibration. Instead of a target with lines, a geometrically flat, white calibration surface is employed. A series of images is taken of the laser plane intersecting the calibration surface with the surface located at incrementally decreasing x positions.

Tilt Angle Calibration

The tilt angle, ϕ_{LP}, defined in Figure 4.2, may be determined by fitting a straight line to the image of the stripe generated from the intersection of the light plane and the calibration surface. A number of images of a sequence of target positions are used for this calibration and the appropriately weighted mean of the slopes of the lines is used (see Figure 6.5). Because the pixel aspect ratio may be different than unity, the slope of the stripe in the image (measured from the pixels) may be slightly different from the slope of the line generated by the light plane

$$\tan \phi_{LP} = r_{pixel} \mathcal{S} \qquad (6.11)$$

where $\mathcal{S}$ is the slope of the stripe measured from the pixels in the image.

Disparity Angle Calibration

The disparity angle, θ_{LP}, is determined by locating the intercept of the stripe in a sequence of images. For convenience, the stripe intercept is taken along the y direction on the $z = z_{COE}$ axis. The intercept information is combined with the test bed x distance settings (this calibration is described in Section 6.3.1) and a least squares slope fitting procedure is used to determine the tangent of the disparity angle (see Figure 6.5).

Light Plane to Camera Distance

This distance, y_{LP} is measured along the y direction of the image plane and is from the center-of-expansion to the light plane. It is obtained from a least squares intercept estimate of the data from the disparity angle calibration (see Figure 6.5).

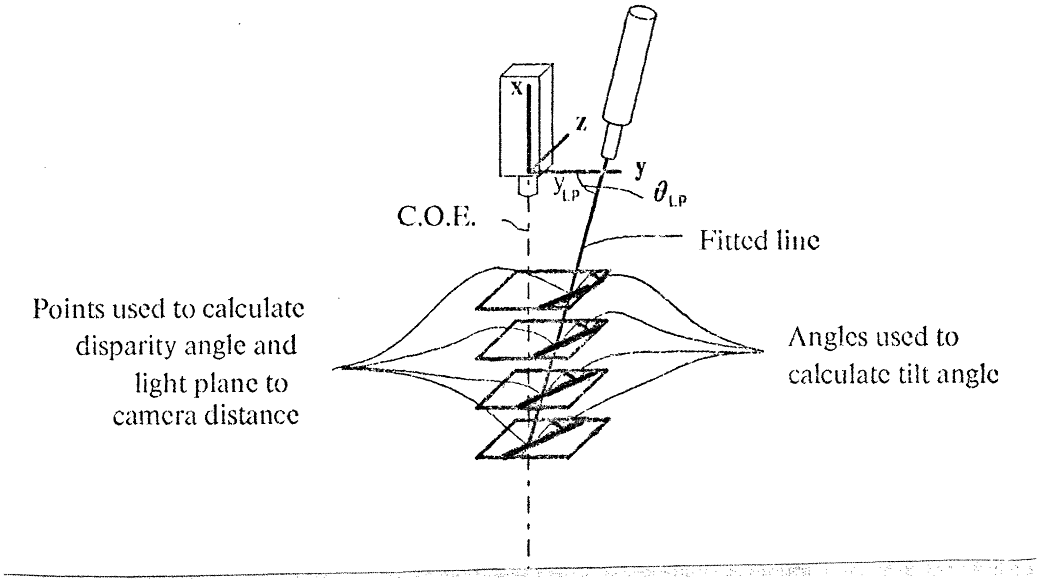

X
Z
y
y_{LP}
θ_{LP}
C.O.E.
Fitted line
Points used to calculate disparity angle and light plane to camera distance
Angles used to calculate tilt angle

Figure 6.5: Method for determining the disparity angle, tilt angle, and light plane to camera distance.

6.4 Performance Evaluation of the Prototype Sensor

The measurement accuracy of the test-bed-mounted sensor was studied by measuring the corner feature of a cube, Figure 6.6. Corner location measurements were compared to the settings on the six stages of the test apparatus. These measurements were relative to an arbitrary reference location. It was not necessary to obtain absolute measurements since relative accuracy is the specification required to determine sensor performance in an assembly environment where all measurements are relative to the robot frame.

The relative accuracy in locating the corner of the cube using information from two light planes was also studied. The maximum likelihood technique developed in Section 4.8 was used to obtain estimates of the orientation of the cube.

6.4.1 Test Procedure

The test procedure for determining the accuracy of the sensor entailed locating the cube at a home position then moving one of the six positioning stages (call this the test direction) a small amount. An image of the cube was recorded and the same stage was moved the same amount once again. This was repeated a number of times. The positions of the cube corner from a series of images were compared to the apparatus positions and sample statistics were calculated from differences between stage motion and sensed corner position. The test procedure was repeated for all six stages. The rotational degrees of freedom in stage motion, x_{rot}, ϕ_{rot}, and ψ_{rot} are defined in Figure 6.6. Measurements were made in the θ, ϕ, and ψ degrees of freedom, which are the Euler angles of the cube. Figure 4.32 shows the convention used. Since the cube was fairly accurately aligned to the axes of the positioner, ϕ_{rot} corresponds closely to ϕ and ψ_{rot} corresponds closely to ψ.

The range of motions for the tests were 0.5 inches (12.7 mm) in each translational degree of freedom and 8.5 degrees (.15 radians) for the ϕ_{rot} rotation and ψ_{rot} rotation and 20 degrees (.35 radians) for the x_{rot} rotation. In general, measurements near the center of the ranges of motion were more accurate than those near the ends. This was because line segments on some cube faces were smaller as the cube was moved from a central position under the laser stripe.

Because the cube is not absolutely referenced to the stationary camera frame, accurate measurements in some degrees of freedom were difficult to obtain. Most measurements were made in degrees of freedom which were stationary during test direction motions. For example, rotation about the x axis slightly changes the y and z positions of the corner of the cube since the rotation is not exactly through the corner, however, the x position remains unchanged. Thus, only estimates of the x coordinate of the corner would be computed.

6.4.2 Single Light Plane Test Results

Figure 6.7 shows a typical plot of the measured location of the cube in the z direction due to motions in the y direction. The larger errors at the ends of the range are due to smaller line segments on some of the cube faces. Similar plots were generated for measurements and motions in all pairs of degrees of freedom. Table 6.1 shows the results of the tests for motions of each of the six degrees of freedom. The blank spaces in the table correspond to degrees of freedom which were non-stationary or were not linearly related to the test direction motion. The mean of the sample standard deviations are .002 inches (.05 mm) in translation and 0.095 degrees (0.0015 radians) in rotation. Assuming the distribution of errors

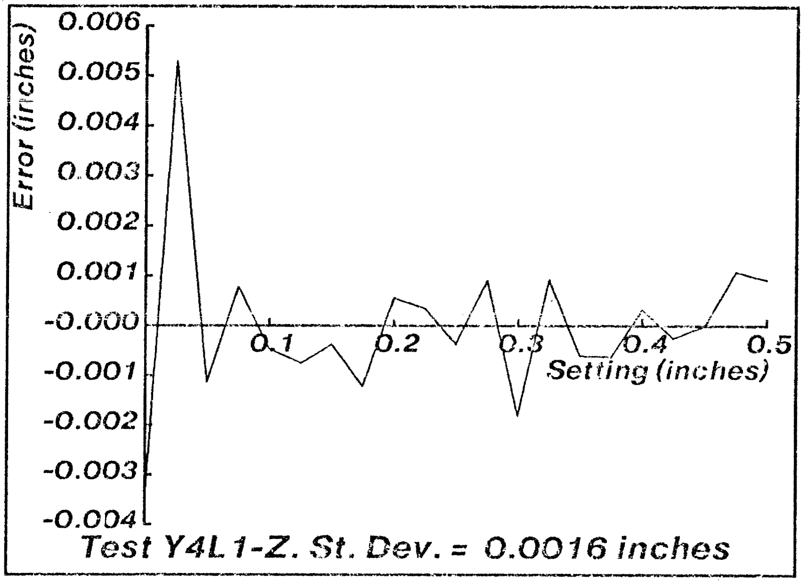


Figure 6.7: Errors in locating the corner of a cube. Measurements were made in the z direction for .5 inches (12 mm) of motion in the y direction.

Direction of Motion	Laser Number	X error (inches)	Y error (inches)	Z error (inches)	θ error (degs.)	ϕ error (degs.)	ψ error (degs.)
Trans. X	1	.0030	.0010	.0010	.05	.03	.08
Trans. X	2		.0025	.0015	.13	.04	.11
Trans. Y	1	.0035	.0025	.0015	.12	.09	.14
Trans. Y	2	.0080	.0025	.0025	.12	.05	.13
Trans. Z	1	.0010	.0020	.0025	.11	.07	.13
Trans. Z	2	.0025	.0005	.0015	.15	.06	.14
Rot. ϕ_{rot}	1			.0015		.05	.08
Rot. ϕ_{rot}	2			.0015		.09	.08
Rot. ψ_{rot}	1		.0020		.09		.12
Rot. ψ_{rot}	2		.0020		.14		.08
Rot. x_{rot}	1	.0015					
Rot. x_{rot}	2	.005					

Table 6.1: Results of the part position sensor accuracy tests. Each number is the sample standard deviation from 21 images. Mean sample standard deviations are: translation – .002 inches (.055 mm), rotation – .095 degrees (.0015 radians).

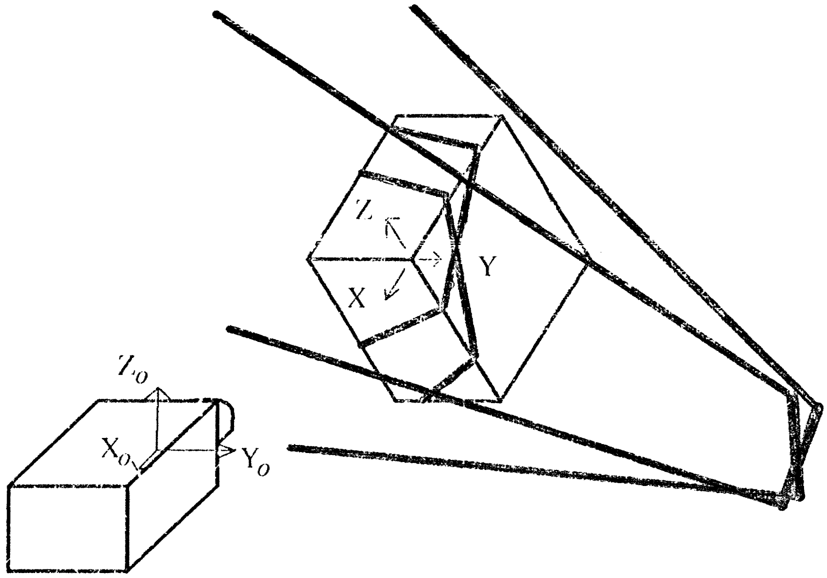


Figure 6.8: The intersection of two independent light planes with a cube.

are Gaussian, 98.8 percent of the measurements (5 standard deviations) will be within an accuracy of .010 inches (.25 mm) and .5 degrees (.007 radians). These are within the design goals presented in Section 3.4.

The repeatability of the sensor was extremely high (about .0001 inches or .0025 mm per standard deviation) when multiple readings were taken without moving the target in between readings. If the target position is perturbed slightly, the readings change more appreciably (about .001 inches or .025 mm per standard deviation).

6.4.3 Multiple Light Plane Test Results

The accuracy in locating the cube's corner using two light stripes was investigated. Six line segments (three from each laser) were generated for each position of the cube (see Figure 6.8). Two independent orientation estimates (Euler angle vectors) were obtained from the line segments. These were combined using a maximum likelihood technique [Equation (4.65)]. Two sets of results were obtained: one with all six estimates being used in a single maximum likelihood calculation, and one with only pairs of corresponding Euler angles being used. When pairs of single

Euler angles are used in the estimate, correlations with other Euler angles are ignored. Figure 6.9 shows the Euler angle estimates for lines from laser number 1, laser number 2, and the maximum likelihood estimate using all six angles. The calculated angles from each of the lasers have different means due to calibration errors. The maximum likelihood estimate does not always fall between the two calculated angles and for some angles seems to be an imprecise estimate. This is most likely due to the coupling between the Euler angles and the unmodeled calibration uncertainties. The maximum likelihood estimates from pairs of angles are shown in Figure 6.10. Here the maximum likelihood estimate is between the two independent laser estimates. The estimate nears one of the single laser estimates when that one is more reliable (longer line segments) than the other. The estimate always uses the combination of the measurements which reduces the overall error. Since calibration errors were not taken into account, the estimate weighs the two calculations equally for equal length line segments. Because the maximum likelihood estimate performs a weighted average of the two estimates, and the two estimates have different means, the sample standard deviation of the maximum likelihood estimate is not always less than the smaller sample standard deviation of the single laser estimates; although, the tests showed that it was usually very close to the lower standard deviation of the two.

6.5 Repeatability and Accuracy of the Unimation PUMA Robot

A study was performed to determine the repeatability and local accuracy of a Unimation PUMA robot using the prototype part position sensor. A right rectangular prism, which was held in the PUMA's grippers, was used as the target for the sensor, Figure 6.11.

6.5.1 Repeatability Test

The test procedure for the robot repeatability test is as follows. Define an arbitrary home position within the active sensing volume for the sensor. Move the arm to various locations around the workspace. After each motion away from the home position return the PUMA to the home position and make a measurement using

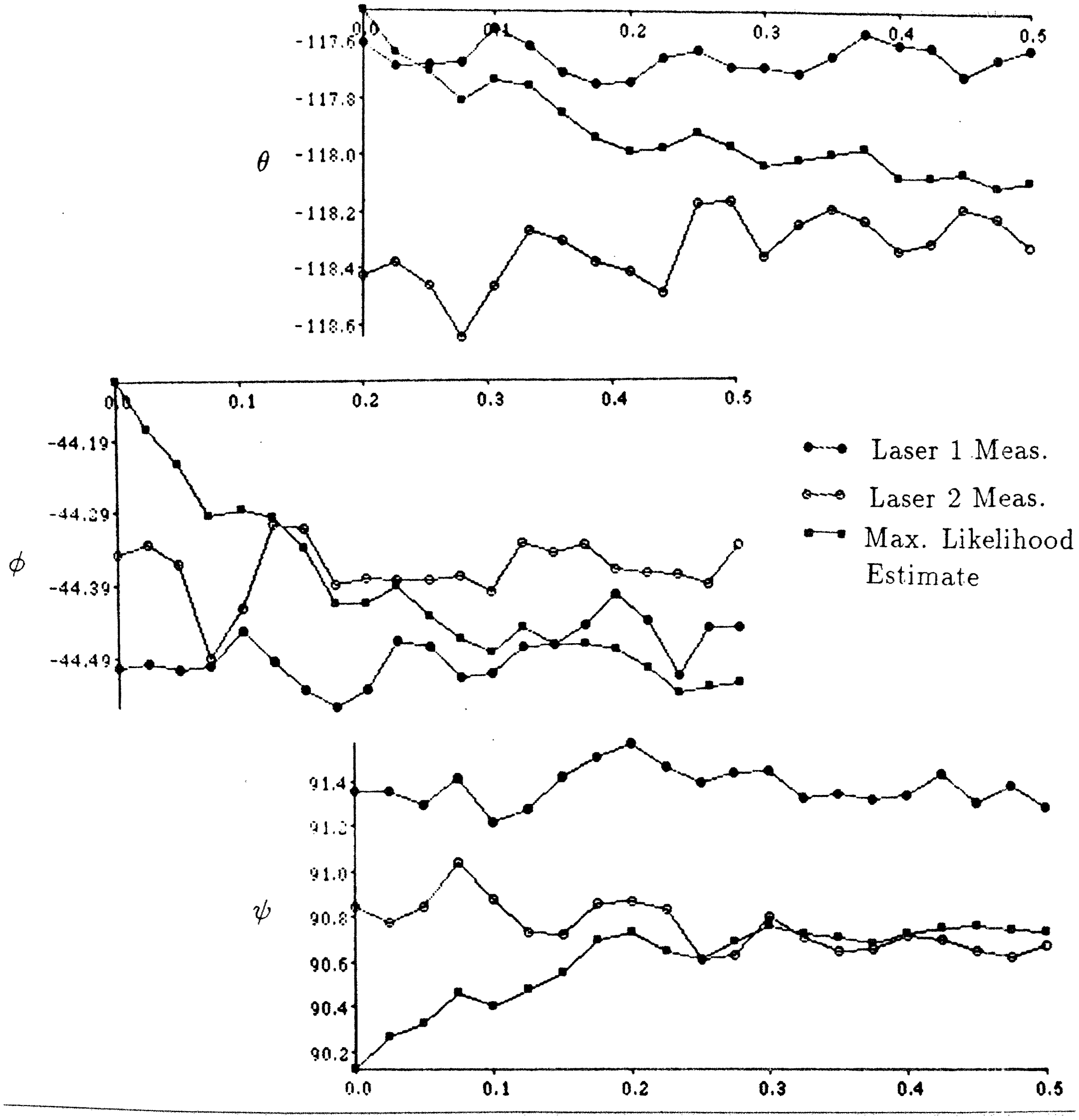


Figure 6.9: Maximum likelihood estimates of rotation angles of the test cube.

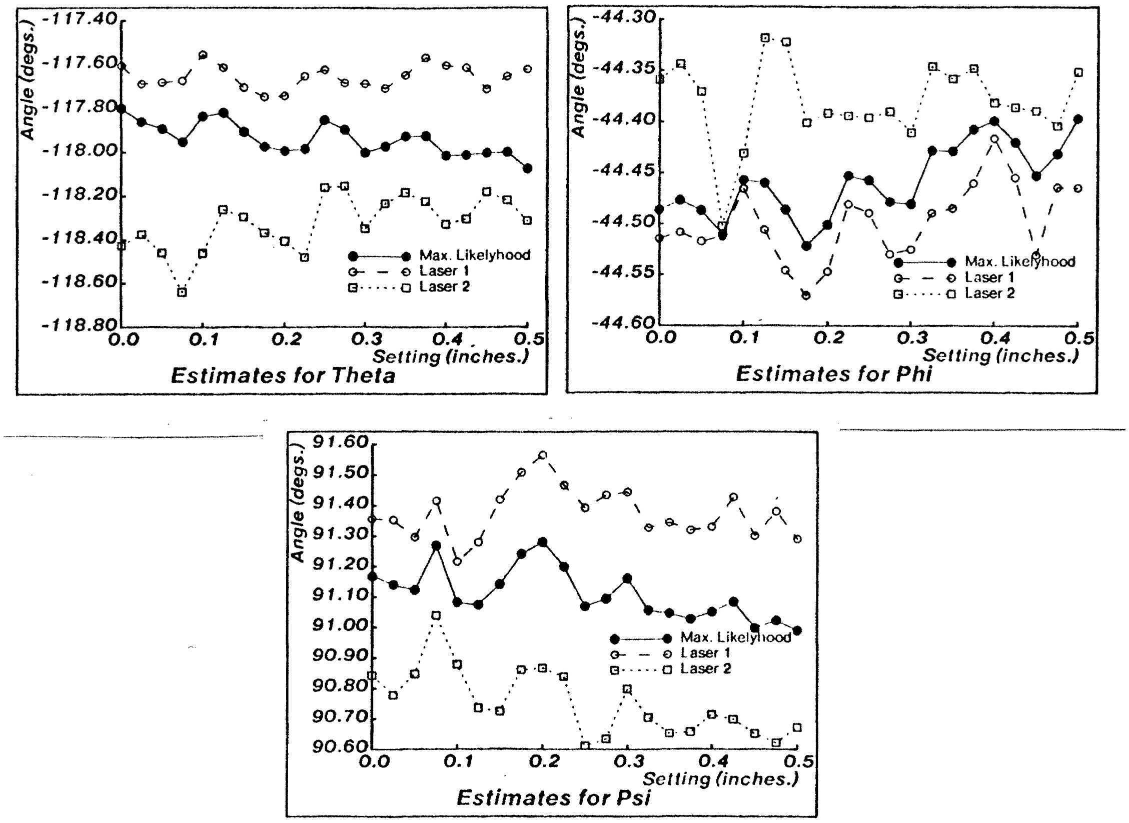


Figure 6.10: Maximum likelihood estimates of rotation angles of test cube from pairs of angles. Plots are for Euler angles θ, ϕ and ψ.

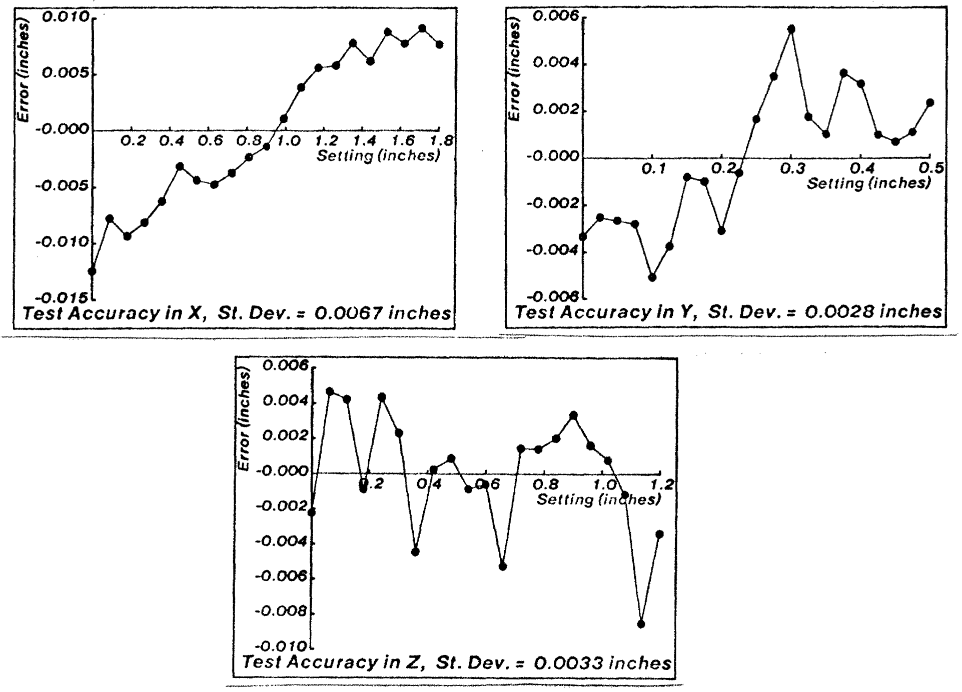


Figure 6.12: Results from the PUMA local accuracy tests. Robot motion in the x, y, and z directions are along the abscissas and measured position deviations are along the ordinates. The standard deviations listed correspond to errors over the full range of motion indicated.

rection parallel to one of its coordinate axes and the sensor was used to measure the total displacement of the prismatic target. Figure 6.12 shows the deviations in PUMA motions as measured by the sensor. As might be expected the local accuracy decreased with larger motions. For instance, the x direction local accuracy over a 1.8 inch (46 mm) range was .0067 inches (.17 mm), whereas, over a .5 inch range in the x direction, the local accuracy was about .003 inches (.08 mm).

6.6 Prototype Sensor-Driven Assembly System

A prototype position-sensor-driven assembly system was developed and its capabilities in performing assembly tasks were studied. The prototype system consists of a Unimation PUMA robot, a part position sensor (described in Section 6.1), a Symbolics Lisp machine model 3600, some image processing hardware, and a color graphics display system (see Figure 6.13). The sensor is mounted on the work table

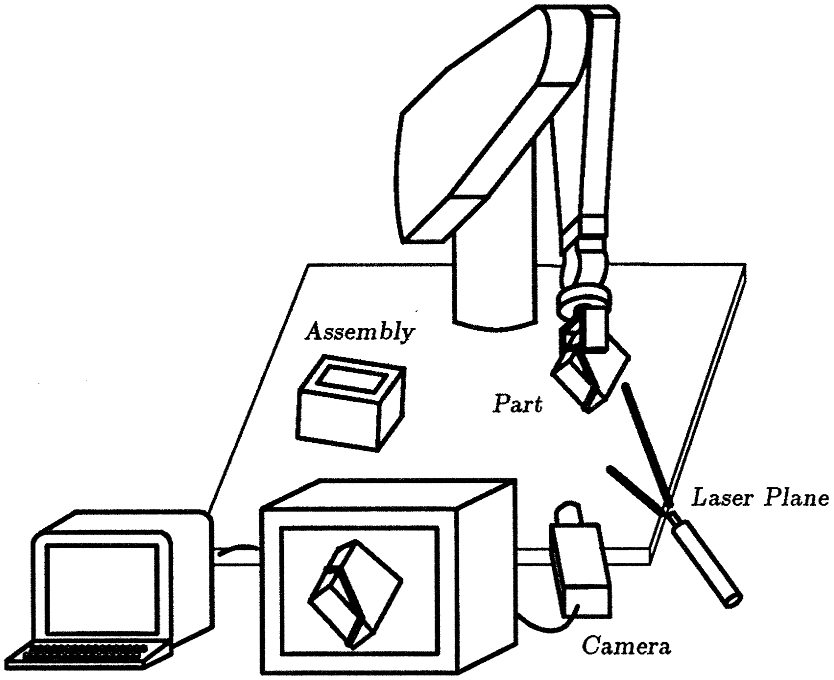


Figure 6.13: The prototype position-sensor-based assembly system. The components include a Unimation PUMA robot, a part position sensor based on a laser-stripe ranging technique, a Symbolics Lisp machine model 3600 and a color graphics display system.

Demonstration Task Procedure

Each of the three demonstration tasks followed essentially the same sensing and assembly procedure, although, the manipulator configurations were often quite different for corresponding phases of the tasks. The manipulator first acquires an approximately positioned part from a prespecified location on the worktable, Figure 6.17a. The part is then held in a sensing location, Figure 6.17b, while the camera grabs a frame, Figure 6.18. After the frame has been grabbed, the manipulator repositions the part to the assembly approach and part realignment position, Figure 6.17c. At this point the manipulator remains idle until the computer has calculated the position of the part from the video frame. After the computer completes the sensing calculation, it displays the results (the location of the corner feature) on the graphics terminal, Figure 6.19. When the part position information is available (sometimes immediately after the robot reaches the assembly approach position), the robot reorients the part and proceeds with the assembly, Figure 6.17d.

Results of the Test Part Assembly

The sensor-driven assembly system was able to reliably assemble all three of the test parts. The processing of the sensed information takes about 2.5 to 3 seconds. This is somewhat dependent on the size of the laser stripes in the image. The sensing volume is located about 24 inches (70 cm) from the assembly jigs; thus, the robot has to make gross motions between sensing and assembly. The results of these test are available on a video tape which accompanies this thesis.

6.6.2 Square Peg-in-hole Analysis and Tests

A three-dimensional square peg-in-hole task was performed using a PUMA robot and the part position sensor. The task was also attempted without the use of the sensor. The rectangular shape is used so that the clearance between the peg and the hole is easily adjusted. The non-axisymetric shape presents some geometric complications [38], but for small misalignments the two dimensional peg-hole-analysis provides a good approximation to the actual physics of the task. The adjustable clearance peg-in-hole apparatus used in the tests in this section is described in

Section 6.6.1.

Theoretical Analysis of the Task

In order to successfully complete the task, both the non-wedging criterion and the chamfer bound criterion, (Equations (2.2) and (2.1) respectively), have to be satisfied. We make the same assumptions and use many of the same quantities as those discussed in Section 2.10. Using Equations (2.40) and (2.41), and the sensor accuracy values given in Section 6.4.2 for ${}^S\sigma_{\epsilon_{F_S}}$ and ${}^S\sigma_{\alpha_{F_S}}$, we may solve for the standard deviations for the total misalignments at the assembly, σ_{ϵ_F} and σ_{α_F}. This results in a translational error of $\sigma_{\epsilon_F} = .006$ inches (.17 mm) and a rotational error of $\sigma_{\alpha_F} = .18$ degrees (.003 radians). For 98.8 percent reliability (5 standard deviations), the chamfers must be

$$L_{chamfer} \geq \frac{5(.006)}{2} = .015 \text{ inches } (.4 \text{ mm})$$

and the clearance ratio must be [from Equation (2.38)]

$$c \geq \frac{5(.3)(.003)}{2} = .0022\ .$$

For a 1.75 inch (44.5 mm) square peg this implies a minimum clearance of .0039 inches (.1 mm).

Experimental Limits of the Task Without Sensing

The peg and hole were carefully aligned then displaced by controlled increments in either rotation or translation. Translation of the peg relative to the hole more than .02 inches (.5 mm) caused the edge of the hole to interfere with the bottom of the peg which caused the insertion to fail. Wedging was initiated by rotating the peg about one of its bottom edges prior to insertion. The depth of insertion before wedging occurred depended on the magnitude of the initial rotation. With the peg misaligned by 2 degrees (no translation error) wedging occurred at a depth of .32 inches (8 mm). With a misalignment of 1 degree, wedging occurred at a depth of .6 inches. Small displacement misalignments did not significantly affect these results. Thus, for the successful assembly of the peg in the hole, the orientation of the peg had to be aligned with the hole's orientation to within about .7 degrees and aligned translationally to within .02 inches (.5 mm).

Experimental Limits of the Task with Sensing

The acceptable initial position of the peg was significantly larger for the system which used the sensor than for the one which did not. At the pickup station the peg could be displaced as much as ±.25 inches (±6.35 mm) in the y direction, ±.75 inches (±19 mm) in the x direction and ±.2 inches (±5 mm) in the z direction. The larger acceptable displacement in the x direction was due to the grippers closing and centering the peg in that direction. The limit in that direction was due to the maximum gripper opening. The volume limitations in the other directions were due to the limited sensing volume of the sensor. Rotational misalignments up to about ±4 degrees (±.07 radians) could be tolerated.

Comparison of Results and Discussion

The error analysis predicted a minimum clearance of .0039 inches (.1 mm). and a minimum chamfer size of .015 inches (.4 mm). The tightest tolerance peg-in-hole assembly which could be reliably performed was with a clearance of .004 inches (.1 mm) and a chamfer of .02 inches (.5 mm). The analysis agrees with the experimental bounds fairly closely. Because the chamfer size on the peg and hole were not easily adjusted, it was difficult to determine if the system could reliably assemble a peg with only .015 inches (.4 mm) of chamfer. Also, not enough data was collected to verify the assumed 98.8 percent reliability.

The system using the part position sensor could assemble parts with initial uncertainties about 10 times greater than the uncertainties in a system without the sensor.

Discussion, Applications, and Conclusions

Chapter 7

7.1 Discussion

Assembly of rigid parts involves the removal of misalignments between them. In order to accomplish this with different shaped parts, a versatile error elimination technique is required. Two methods for eliminating position errors and performing mechanical assembly tasks are presented in this thesis: the *a priori suppression* technique and the *measurement and elimination* technique. The *a priori suppression* technique uses primarily passive methods to control the propagation of position errors; whereas, the *measurement and elimination* technique uses sensing and manipulation to reduce interpart alignment errors. Technical issues for both techniques were dealt with, but the majority of the work concentrated on developing the less traditional error *measurement and removal* technique. In this section, further developments which are required for flexible assembly systems so that they may be used in the factory are discussed.

7.1.1 Development of Computer Integrated Manufacturing Systems

This thesis addresses only a small part of the computer integrated manufacturing problem. Many unresolved problems in areas such as product design for automated assembly, material handling, flexible parts feeding, workcell design, plant layout, and production scheduling have yet to be resolved before an efficient and workable flexible automated factory may be built.

In order for flexible assembly systems to handle larger classes of operations, new tools in addition to those described in Chapter 5 must be developed. In conjunction with the development of tools, new assembly techniques should be developed. For example, there is a need for fasteners which may be easily inserted by a mechanical manipulator and would take the place of machine screws, nuts, and rivets.

Parts feeding and orienting operations are not directly addressed in either of the prototype systems presented in Chapters 5 and 6. There is a need for feeding systems that are flexible and reliable; that is, systems which may be easily reconfigured (preferably automatically) to feed and orient many different parts. Because standard parts feeding equipment is customized for particular parts and, therefore, fairly expensive, it is often not economic to purchase separate feeding devices for each part unless they will be fully utilized (constantly feeding parts for two or three shifts). Programmable feeders which may be reconfigured to feed different shaped parts would be useful in an industrial assembly system. Such feeding systems would be more highly utilized since many different parts could be run through them. Feeding similarly shaped parts with a certain feeder would facilitate the development of flexible feeding systems. It is easier to develop separate systems to feed disc, rod and prismatic shaped parts than a single system which can feed all of them.

One problem with many present-day robot systems is that they perform tasks slow relative to the speed with which humans perform similar tasks. Since man-made mechanical devices are not biologically bound in either speed or force, there is no *a priori* reason that they should run so slow. Insufficient speed is often the factor which makes these systems uneconomic in many industrial applications. Sensing and control systems which are used in an industrial environment must not contribute significantly to the time required to assemble a product. The only factors which should limit a manipulator's speed are inertial, actuator power, and part frailty.

7.1.2 Development of the Sensor

Although the part-position sensing system presented in Chapter 4 takes a few seconds to make a measurement, slight changes will enable it to make measurements in about .5 seconds. The required changes entail a faster hardware convolver and possibly other electronic hardware which can scan through the image and extract the intensity values of the pixels of interest. Both of these hardware implemented features have recently become commercially available [48,96].

More accurate information may be obtained from the sensor if multiple measurements are taken. Because of quantization effects, it is necessary to slightly shift the object with respect to the imaging device in order to take advantage of the multiple measurements. If the object is not shifted with respect to the discrete array, the same errors in position measurements are obtained and the noise cannot be filtered. A lower resolution camera or a larger field of view may be used if the measurement environment allows multiple readings to be taken. Section 4.5 gives the technique for combining multiple measurements. If the images used for the measurements are almost the same, the technique degenerates to a simple averaging technique.

Multiple images or single images with multiple light source planes may be used to sense features which are difficult to locate from an image of the intersection between the object and a single light plane. Objects such as circuit boards, and conical and cylindrical features may be accurately located by intersecting them with multiple light planes positioned in very different locations in space. Alternatively, the objects may be moved in between subsequent images of intersections with a single light plane in order to obtain different "cuts" through the surfaces of the object.

The technique for optimal scaling of each measurement of a feature was developed for straight line image features. Non-polyhedral features generate non-linear curves of intersection and a method of determining the covariance matrix for the parameters of more general curves is required.

More work is required in sensing non-polyhedral features and analyzing nonlinear curves in light stripe images. Some researchers have reported difficulty in fitting conic sections to data [26]. Multiple light plane intersection of cylindrical features produce additional constraints which could be used in the fitting procedure to produce better results.

7.1.3 Development of Position-Sensor-Based Assembly Systems

In the sensor-based assembly system described in Chapter 6, it is necessary to let the manipulator come to rest before sensing the position of the part. The overall cycle time of the system may be reduced by sensing the part while the robot is moving. If the sensor is mounted off board the moving robot, a frame grabbed with a video camera produces a blurred image. The part location can be "frozen" if the light source is strobed at a high power and short pulse width. Either a pulsed laser or a fast mechanical shutter may be used. In addition, the light pulse must be timed such that it occurs during the time that the pixels are integrating the impinging radiation and not while they are shifting out their accumulated output. In order for the system to know the location of the part when the robot finally comes to rest, the location of the manipulator must be known at the same moment that the light pulse occurs (the instant the measurement is taken). Immediate joint encoder data access (manipulator position) is required. Even if the position of the robot is known at the time of measurement, dynamic forces and torques may produce an offset in the predicted absolute position of the part; although, it may be possible to take the deflections due to dynamic loads into account when calculating the part's position. If the sensor is mounted on board the moving robot, it might be possible to freeze all degrees of freedom between the sensor and the part and make an accurate measurement with a moving robot. The manipulator joints are available for either gross motion or positioning of the part in the sensor field of view and with only six joints, it might be difficult to both move the part and the sensor in unison and have enough degrees of freedom in part positioning to obtain a good view for measurement.

Using position sensing data to calculate manipulator position commands allows certain components of the system to be somewhat positionally sloppy. The pallets, or feeding-orienting devices need not locate the parts precisely. A system which constrains the general orientation of a part is much less expensive and more flexible than a system which must constrain the part in some absolute location. The robot grippers also do not need to precisely locate the part, although, they must firmly hold the part with respect to the last link of the manipulator. In this case "firmly" means some compliance is acceptable, but slip is not.

7.2 Applications of the Sensing Technology

The light stripe sensor presented in this thesis has many applications in addition to supplying part position information to a mechanical assembly system. Light stripe systems in the literature (Section 3.1) independently use each of the points along the light stripe and usually no curve is fit through sets of points. Fitting a curve through the points in the image, then calculating the location of the feature of interest in three-space is faster than projecting each point into three-space then fitting them to the feature. The resulting higher speed sensing system has uses in locating part surfaces during closed-loop machining operations, and inspecting part surfaces after being machined. The system is also useful in determining whether parts are in the proper orientation in a flexible feeding system. Either the feeder could recirculate parts which are not in the proper orientation, or the information could be passed to the manipulator which would compensate for the displaced part by altering the grasp position.

Since the sensor system makes accurate six degree-of-freedom measurements it may be used as a robot calibration system. Robot accuracy errors occur because the position of the manipulator does not correspond to the commanded position. Using a sensing system to measure the position of the manipulator, correction factors may be calculated such that for a given position command, the repeatable errors are eliminated and the robot moves very near the commanded position. The sensing system described in this thesis does not have sufficient resolution to make accurate measurements in the large field of view corresponding to an entire robot workspace. Multiple sensors or a single sensor whose field of view may be aimed at different sections of the workspace could be used or a higher resolution camera could be used. Section 3.3 lists other uses for the sensor in an industrial environment.

The sensing system presented in this thesis gives useful information about the location of objects only if they are very near where the system expects them to be. The system cannot deal with randomly oriented objects. In order to determine an object's location without *a priori* orientation information, a system such as the sparse data system described in [76,77,78,79] could be used as a preprocessor. The part could then be grasped and realigned so that the light stripe sensor could make an accurate measurement.

Instead of measuring the location of the part remote from its final destination, the sensor could be used to measure the location of the part very near its final position. A position loop could then be closed around the manipulator which would allow the part to be positioned more accurately. Errors due to uncertainties in robot motion and sensor-robot calibration discussed in Section 2.7.7 would be virtually eliminated.

7.3 Technical Contributions and Conclusions

This thesis has addressed many areas related to flexible assembly systems and precision measuring techniques; however, the major contribution of the work is the analysis, development and demonstration of a position-sensor-driven assembly system. The success of the system was due to the task requirement and error analysis studies for mechanical assembly operations, the development of the laser stripe vision system, and the algorithms associated with extracting accurate position information from noisy images. The position error measurement and removal technique employed by the sensor-based assembly system provides high flexibility for handling many different parts without custom fixture designs. Because the system eliminates alignment errors before the mating phase of an assembly and does not attempt to monitor the assembly operation during the mating phase, parts mating progresses faster than it would otherwise if limited-bandwidth force feedback were used. Mating occurs open loop and relies on the passive compliance of the system to eliminate any alignment errors remaining after the measurement and removal phase. Because the sensor-robot system can eliminate most of the alignment errors, mating is less sensitive to the passive compliance parameters. The system was able to accomplish assembly operations with initial part misalignments 5 to 10 times greater than allowable misalignments in a system which does not use a sensor.

The prototype sensing system used to make the part position measurements was relatively inexpensive and operated with sufficiently high accuracy and speed. Theoretical and experimental studies showed that the prototype sensor coupled with a commercial six degree of freedom manipulator could perform peg-in-hole type assembly operations on parts with clearance ratios (clearance/diameter) as low as about .0025. This clearance ratio corresponds to many high precision industrial assembly tasks.

In addition to the development of the technology for position-sensor-based assembly this thesis presents a technique for analyzing the potential success of this type of system. Chapter 2 enumerated the various error sources which occur in assembly systems and presented a methodology for combining the errors and determining the feasibility of an assembly task given a certain set of resources. The example error calculation for the square peg-in-hole task gave results which were verified through experiments in Chapter 6.

In order to make accurate measurements with a discrete element sensor, it is necessary to apply certain filtering techniques to the image. A number of techniques for finding the location of a light stripe on a row by row basis were presented in Chapter 4. It was found that the light stripe feature could be located to subpixel resolution along each row of the image. The light stripe location on each row was combined to find the parameters of a least squares fitted line. The accuracy of finding the parameters of fitted lines was also discussed and found to be highly dependent of the length of the image of the light stripe.

In addition to using certain image processing techniques, careful sensor system calibration was important for making accurate position measurements. A relatively fast and simple procedure was developed for performing accurate camera and laser-stripe calibration.

Techniques for locating objects from sparse light stripe data were central to the success of the real-time-part-position sensor. Because a closed-form solution to the right-corner-feature location problem was used, the computation time for locating these features was very short.

The selection of the ranging technique was based on a literature review of ranging technologies and a study of the accuracy of two feature locating techniques (Chapter 3). The light stripe ranging technique was selected because of its potential for high resolution, fast operation, and relative simplicity in making three-dimensional measurements. The feature locating technique was chosen over the fitted boundary interpolation technique because of its potential for high accuracy and relative simplicity.

A technique for increasing the accuracy of position measurements by combining multiple measurements was presented in Chapter 4. The technique optimally combined object position estimates based on the expected accuracy of each measurement. The technique can be used on position estimates obtained from different sources so long as an estimate for the quality (covariance matrix) of the measurement is available.

In addition to being used as an assembly system tool, the sensor was used as a robot position measuring device. Repeatability and local accuracy measurements of a Unimation PUMA 600 manipulator were made. The measured specifications were then used in error calculations in Chapter 2.

Other contributions in this thesis include the development of mechanically programmable tools to address certain difficult assembly operations. The need for some of the tools were determined from an assembly task analysis. The tools were designed so that a robot could easily reconfigure them to fit its needs for a particular assembly task. The tools were integrated into a prototype flexible assembly cell and used to assemble consumer hand drills.

Experiments showed that the position-sensor-based assembly system can make more accurate non-contact part misalignment measurements and corrections than unaided humans. With more accurately aligned parts, failure modes such as wedging, jamming and missed chamfer crossing are less likely to occur. This means that the process requires less real-time monitoring and can proceed relatively fast. Thus, position-sensor-based machine assembly can be more efficient than manual assembly.

Path System Definitions

Appendix A

A.1 Paths

A path is defined as an ordered set of N points (P_i, $i = 1, \ldots, N$) in 3-space (see Figure A.1). The path coordinate system, with axes X_P, Y_P, Z_P and corresponding unit vectors $\hat{x}_P$, $\hat{y}_P$, $\hat{z}_P$ are defined by the first, and last points of the path in the following manner. The X_P axis of the path coordinate system is along the vector from the first point of the path to the last point. The Z_P axis of the path coordinate frame is positioned such that the global Z_O component of the unit $\hat{z}_P$ vector is maximized (or the X_P, Z_P plane is made orthogonal to the X_O, Y_O plane). If the X_P axis of the path coordinate system is parallel to the global Z_O axis, the Z_P axis of the path coordinate system is made parallel to the global Y_O axis. The two approach points of the path (the second and second to last points) must always have the same X_P and Y_P coordinates as their respective end points.

The path may equivalently be described by a set of vectors $\vec{r}^{P}_{P,i}$ whose X_P, Y_P, Z_P components are the X_P, Y_P, Z_P coordinates of the corresponding set of points. The vector notation $\vec{r}^{A}_{B,i}$ describes a vector from the origin of the A coordinate system to the i^{th} point in the B coordinate system.

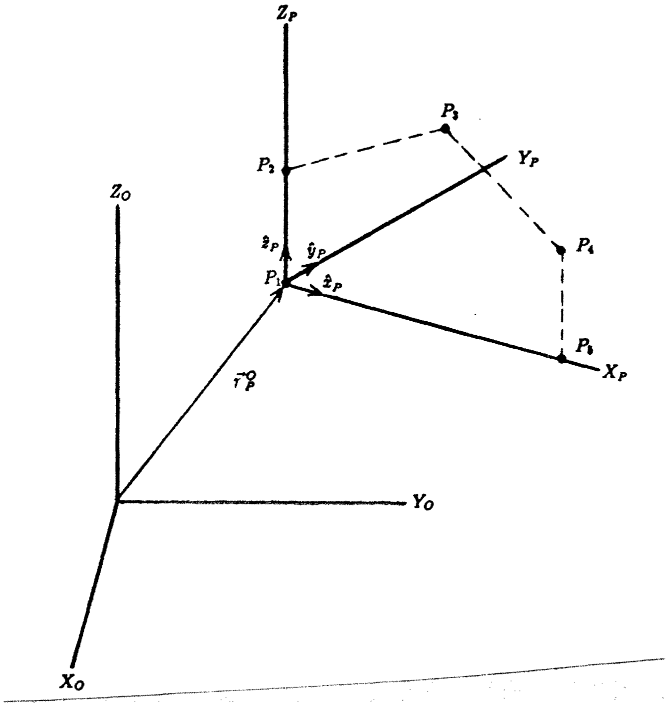


Figure A.1: Path coordinate system.

A.2 Path Transforms

A.2.1 Spatial Path Transforms

The spatial path transform translates, rotates and scales a predefined path to the first and last trajectory points defined in the global X_O, Y_O, Z_O coordinate system. The trajectory coordinate system is defined by the X_T, Y_T, Z_T axes and the $\hat{x}_T$, $\hat{y}_T$, and $\hat{z}_T$ unit vectors which are located by the position of the start and end points using the same technique described for the path coordinate system (see Figure A.2). The intermediate points of the trajectory are located in the trajectory frame by the scaled path vectors. The scaling factor for the vectors of a path is the ratio of distances from the first to last trajectory points and the first to last path points. For a path and a corresponding trajectory with N points, the scale factor is:

$$S.F. = \frac{\left|\vec{r}^{\,T}_{T,N}\right|}{\left|\vec{r}^{\,P}_{P,N}\right|} \tag{A.1}$$

where the P and T super- and subscripts denote references to the path and trajectory coordinate frames respectively. To scale the path, each $\vec{r}^{\,P}_{P,i}$ (the i^{th} path

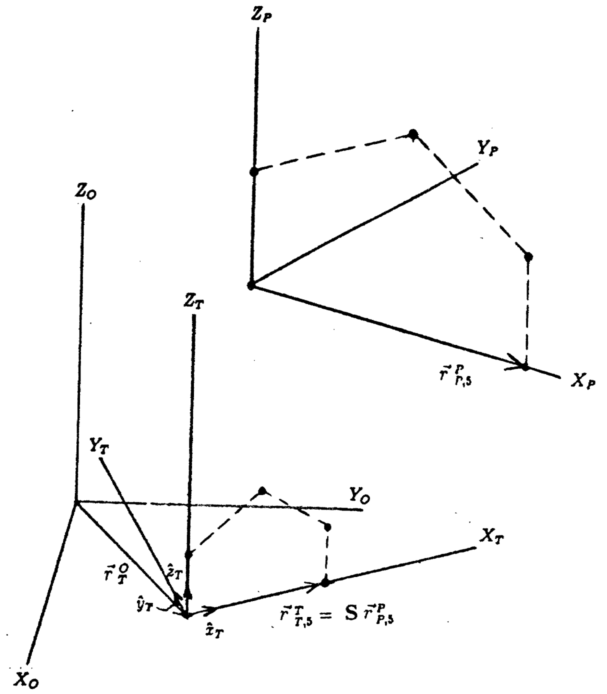


Figure A.2: The spatial path transform.

position vector) may be multiplied by a scaling matrix.

$$\mathbf{S} = \mathbf{I}\; S.F. \tag{A.2}$$

where **I** is the 3 X 3 identity matrix. Since the first point of the path is at the origin, the path scaling creates no offset from the path coordinate system.

The path and trajectory coordinate frames were defined in similar fashions (from the two end points); thus, the scaled path vectors, $\mathbf{S}\vec{r}^{\,P}_{P,i}$, may be used as the X_T, Y_T, Z_T coordinate system vectors for the trajectory. Since the robot controller uses joint angles (global coordinates) to move the robot, the trajectory vectors must be expressed in the global frame. The set of transformed vectors from the path coordinate frame to the global frame with end points at the specified start and end trajectory points is:

$$\vec{r}^{\,O}_{T,i} = \vec{r}^{\,O}_T + \mathbf{R}_{T,O}\,\mathbf{S}\,\vec{r}_{P,i}, \qquad i = 1, \ldots, N \tag{A.3}$$

where $\vec{r}^{\,O}_T$ is the vector from the origin of the X_O, Y_O, Z_O frame to the origin of the X_T, Y_T, Z_T frame, and $\mathbf{R}_{T,O}$ is the 3 X 3 rotation matrix from the X_T, Y_T, Z_T frame to the X_O, Y_O, Z_O frame:

$$\mathbf{R}_{T,O} = [\hat{x}_T\ \hat{y}_T\ \hat{z}_T] \tag{A.4}$$

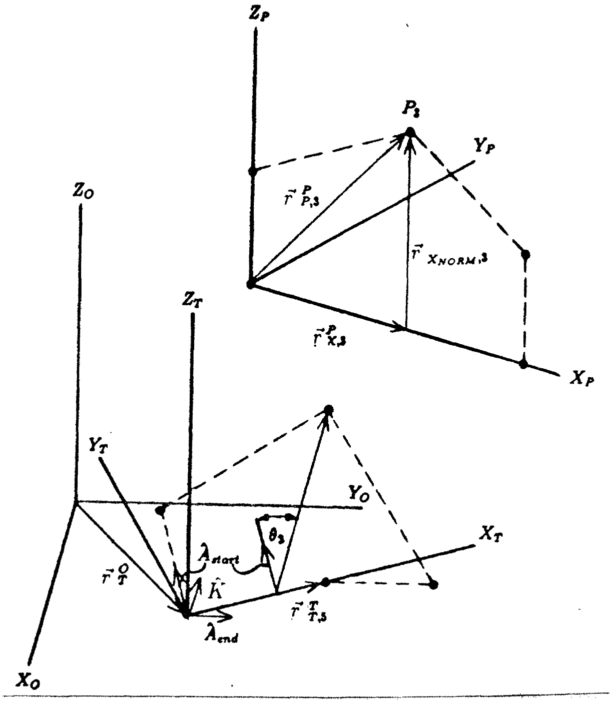


Figure A.3: The cylindrical path transform.

A.2.2 Cylindrical Path Transforms

The cylindrical path transform warps the original path space to a specified trajectory space. The path coordinate system and the trajectory coordinate system are defined as above. In addition to the two end points, a trajectory specification includes starting and ending approach unit vectors. These approach vectors specify the direction of the robot departure from the start point and the direction from which the robot should approach the end point. In a cylindrical transformation, each of the vectors to the path points in the path coordinate frame are scaled and rotated as a function of their X_P path coordinate. The cylindrical path scaling is similar to the spatial path transform scaling except that it is performed only on the path X_P coordinate of each path vector. For a cylindrical path transform, it is necessary to split each path vector, $\vec{r}^{\,P}_{P,i}$, into two components, $\vec{r}^{\,P}_{X,i}$ and $\vec{r}_{X_{NORM},i}$. The $\vec{r}^{\,P}_{X,i}$ vector lies along the X_P axis and the $\vec{r}_{X_{NORM},i}$ vector is orthogonal to the X_P axis and terminates at P_i (see Figure A.3).

$$\vec{r}_{X_{NORM},i} = \vec{r}^{\,P}_{P,i} - \vec{r}^{\,P}_{X,i} \tag{A.5}$$

The action of the cylindrical path transform scales the $\vec{r}^{P}_{X,i}$ vectors and rotates the $\vec{r}_{X_{NORM},i}$ vectors. The scale factor is the same as that in Equation A.1. The $\vec{r}_{X_{NORM},i}$ vectors are rotated about an axis parallel to a vector, $\vec{k_A}$, which is orthogonal to the start and end approach unit vectors ($\hat{A}_{start}$ and $\hat{A}_{end}$ - defined in the trajectory frame)

$$\vec{k_A} = \hat{A}_{start} \times \hat{A}_{end} \tag{A.6}$$

where × is the vector product operator. The angle of rotation for the $\vec{r}_{X_{NORM},i}$ vectors is a fraction of the total rotation angle, θ, between the two approach vectors. θ may be found by satisfying the following two equations (from the definition of the vector scalar product and the magnitude of the cross product) in the interval $[0, \pi]$:

$$\theta = cos^{-1}(\hat{A}_{start} \cdot \hat{A}_{end}) \tag{A.7}$$

$$\theta = sin^{-1}(|\vec{k_A}|) \tag{A.8}$$

In the cylindrical transformation, each $\vec{r}_{X_{NORM},i}$ vector is first rotated via a rotation matrix, $\mathbf{R}_{T,A}$, which rotates the $\hat{z}_T$ vector into the $\hat{A}_{start}$ vector and then by a rotation matrix $\mathbf{R}_{K,i}$ which rotates the $\hat{A}_{start}$ vector by an angle θ_i towards the $\hat{A}_{end}$ vector. The expression for a rotation matrix $\mathbf{R}$ which rotates a vector about a rotation vector $\hat{K} = [k_x k_y k_z]^T$ and through an angle ϕ is

$$\mathbf{R} = \begin{bmatrix} k_x k_x vers(\phi) + cos(\phi) & k_y k_x vers(\phi) - k_z sin(\phi) & k_z k_x vers(\phi) + k_y sin(\phi) \\ k_x k_y vers(\phi) + k_z sin(\phi) & k_y k_y vers(\phi) + cos(\phi) & k_z k_y vers(\phi) - k_x sin(\phi) \\ k_x k_z vers(\phi) - k_y sin(\phi) & k_y k_z vers(\phi) + k_x sin(\phi) & k_z k_z vers(\phi) + cos(\phi) \end{bmatrix} \tag{A.9}$$

where $vers(\phi) = 1 - cos(\phi)$ (from [147]).

Thus, the expression for the $\mathbf{R}_{T,A}$ matrix is Equation A.9 with $\hat{K}$ and ϕ set to

$$\hat{K} = \frac{\vec{k}_{T,A}}{|\vec{k}_{T,A}|}$$

$$\vec{k}_{T,A} = \hat{z}_T X \hat{A}_{start}$$

$$\phi = cos^{-1}(\hat{z}_T \cdot \hat{A}_{start}) = sin^{-1}(|\vec{k}_{T,A}|), \quad \phi \in [0, \pi]$$

For the $\mathbf{R}_{A,i}$ matrix, $\hat{K}$ and ϕ are

$$\hat{K} = \frac{\vec{k_A}}{|\vec{k_A}|}$$

$$\phi = \theta_i = \theta \ \frac{|\vec{r}^{P}_{X,i}|}{|\vec{r}^{P}_{P,N}|}$$

The cylindrically transformed trajectory vectors with respect to the global frame are

$$\vec{r}^{O}_{T,i} = \vec{r}^{O}_{T} + \mathbf{R}_{T,O} \left(\mathbf{S} \vec{r}^{P}_{x,i} + \mathbf{R}_{A,i} \mathbf{R}_{T,A} \vec{r}_{X_{NORM},i} \right), \quad i = 1, \ldots, N \tag{A.10}$$

7565 Workspace Waviness

Appendix B

Prior to positioning the reference baseplates (Section 5.2.1) onto the 7565's work surface, the variation of the 7565 surface plate height was measured. For assembly task programming, the surface height variation with respect to the manipulator reference frame (as an observer in the gripper reference frame would measure the surface) is more critical than absolute height variation. The changes in height with respect to the manipulator frame was measured using a mechanical indicator. The indicator was placed in the grippers and Z direction measurements were taken every 2 inches over the entire X-Y workspace, Figure B.1. The resulting indicator readings are shown in two perspectives in Figure B.2. The low point of the surface is 4 inches forward and 2 inches right of the left rear corner in the first view of Figure B.2. The highest deviation measured occurs at the near right corner of the same view and is .012 inches above the lowest point.

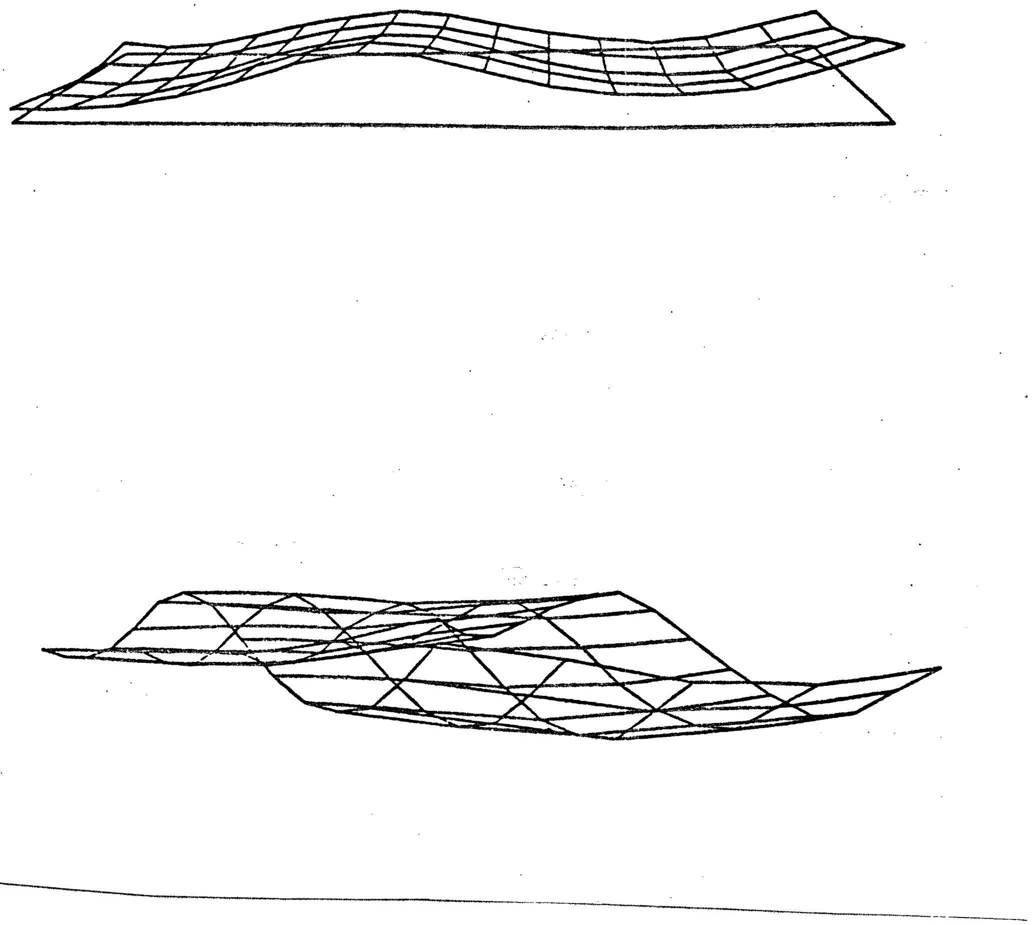


Figure B.2: Two views of 7565 surface plate height variation.

Flexible Fixture Design Calculations

Appendix C

The flexible fixtures in the prototype assembly cell are locked into place by tightening a cable in the arm of the device, Figure C.1. The arm consists of alternating spheres and cylinders with holes through the center and a steel cable running through the holes. In this section, the relationship between cable tension and the maximum allowable force applied to the end of the device without ball/cylinder slip is determined.

A free body diagram of one of the balls of the arm is shown in Figure C.2. We assume that the slip is brought about by the "bending" moment applied to the ball from an applied force at the end of the arm

$$M_B = F_{app}x$$

where M_B is the moment applied to the ball, F_{app} is the force applied to the end of the flexible fixture and x is the distance form the end of the fixture to the center of the ball. We assume that the shear force across the arm does not make a significant contribution to the forces at the ball/cylinder interface. The cylinder radius at the point of contact with the sphere is R_C and the sphere radius is R_S. The angle of contact, θ, is defined by the equation

$$\cos\theta = \frac{R_C}{R_S}.$$

ϕ is the angle about the contact circle with $\phi = 0$ pointing towards the applied force F_{app}.

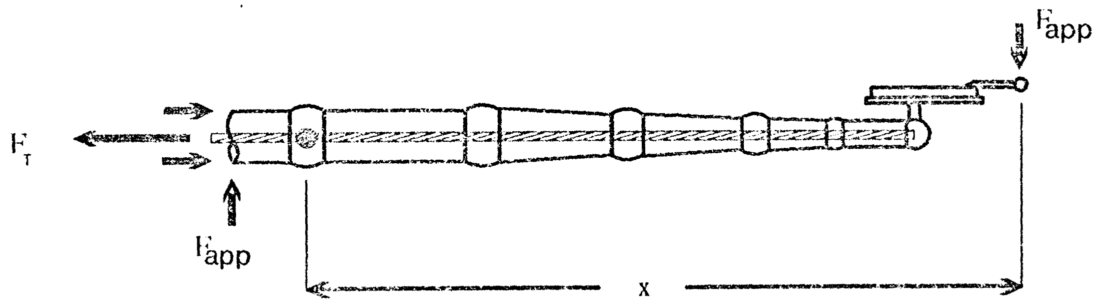


Figure C.1: Structure of the flexible fixture arm.

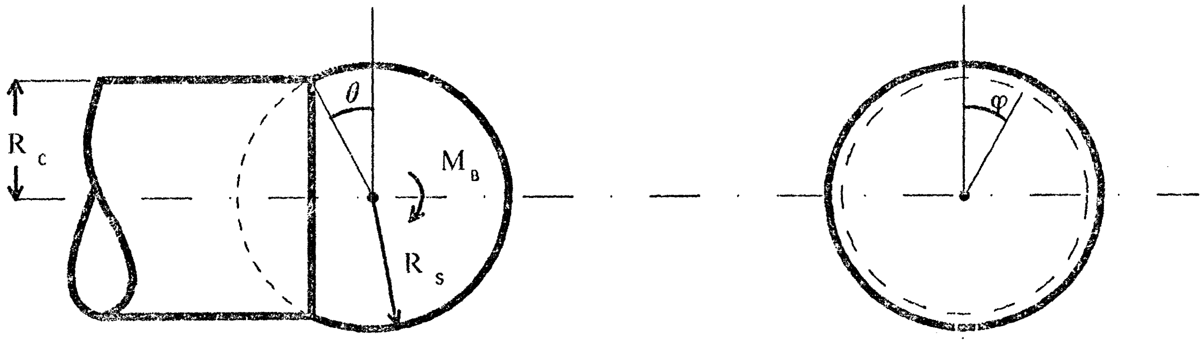


Figure C.2: Free body diagram of a single ball in the fixture arm.

The normal force per unit length along the contact circle is

$$\sigma_N = \frac{F_T}{2\pi R_C \sin\theta} = \frac{F_T}{2\pi R_S \sin\theta\cos\theta}. \tag{C.1}$$

where F_T is the force pushing the ball and cylinder together along the cylinder axis. If the sphere slips in the cylinder, points on the cylinder edge would trace out circles (lines of latitude about a horizontal axis) on the surface of the sphere. The frictional force per unit length of contact is directed tangent to these circles and has magnitude

$$\sigma_F = \mu\sigma_N \tag{C.2}$$

where μ is the coefficient of friction. The distance from a point of contact to the sphere center perpendicular to the axis of potential rotation is

$$R_{SM} = \sqrt{R_S^2 + R_C^2\cos^2\phi - R_C^2} = R_C\sqrt{\frac{1}{\cos^2\theta} - \sin^2\phi}. \tag{C.3}$$

The torque contribution along a differential segment of the contact circle is

$$dM = R_{SM}\sigma_F(R_C d\phi). \tag{C.4}$$

Substituting Equations (C.1), (C.2), and (C.3) into Equation (C.4) and integrating around the contact circle gives the total moment due to friction which can resist slipping

$$\begin{aligned} M &= \frac{\mu R_S F_T}{\pi\sin\theta}\int_0^\pi \sqrt{1-\cos^2\theta\sin^2\phi}\, d\phi \\ &= \frac{2\mu R_S F_T}{\pi\sin\theta}E\left(\cos\theta, \frac{\pi}{2}\right) \end{aligned} \tag{C.5}$$

where E is the solution to an elliptic integral of the second kind. The function $log\left[\frac{2}{\pi\sin\theta}E\left(\cos\theta, \frac{\pi}{2}\right)\right]$ is plotted for θ from 0 to $\frac{\pi}{2}$ in Figure C.3.

To maximize the resisting moment the contact angle should be small; however, too small a contact angle will produce a locked condition between the ball and cylinder. In order for the structure to remain unlocked we need $\theta > \tan^{-1}\mu$. Assuming steel on steel with a coefficient of friction $\mu = .3$, the value of the contact angle which just produces locking is $\theta = .29$. Reading the scaled moment from Figure C.3, the maximum resisting moment just before the locking condition is

$$M_{max} = \log^{-1}(.37)\mu F_T R_S = .7 F_T R_S$$

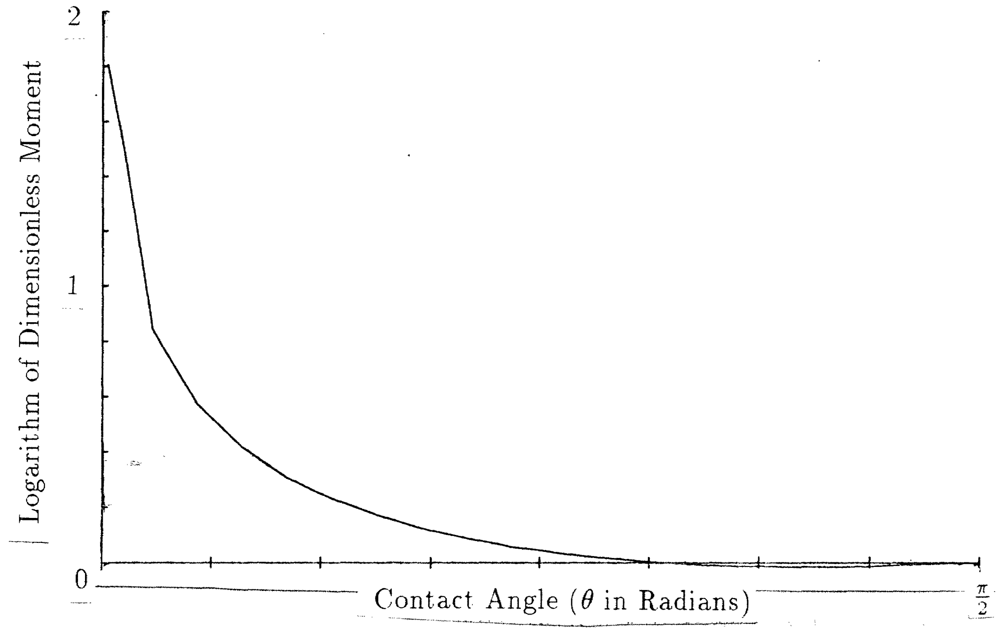


Figure C.3: Log plot of scaled resisting moment as a function of the ball-cylinder contact angle for the flexible fixture arm.

and the maximum allowable applied force is

$$F_{app,max} = .7F_T\frac{R_S}{x}.$$

For example, given a steel sphere with radius $R_S = .5$ inch (12.7 mm) and a cable with tension $F_T = 700$ lbs. (3114 Nt.), the maximum force which may be applied at 10 inches (254 mm) from the ball is 24.5 lbs. (109 Nt.).

Errors Generated with a Spherical Wrist

Appendix D

In this section we investigate errors in positioning the endpoint of a spherical wrist joint, Figure D.1, due to errors in actuator positioning. The different types of endpoint positioning errors for robot manipulators are discussed in Section 2.7. Errors may be divided into repeatable errors and nonrepeatable errors. A nonrepeatable error (probabilistic) is essentially the repeatability (as defined in Section 2.4) at a point in the workspace. The repeatable error is essentially the accuracy minus the repeatability. We assume that errors are either due to inaccuracies in joint positioning or to other factors such as non-rigid members, link geometric inaccuracies and misalignment, etc. The repeatable and nonrepeatable errors in joint positions produce errors at the endpoint proportional to the value of the Jacobian at that particular configuration. The four components of endpoint error are

$$\vec{e}_{endpoint} = \mathbf{J}(\vec{x})[\vec{e}_{rep}(\vec{x}) + \vec{e}_{r\bar{e}p}(\vec{x})] + \vec{f}_{rep}(\vec{x}) + \vec{f}_{r\bar{e}p}(\vec{x}) \qquad \text{(D.1)}$$

where x is the vector of joint positions, $\mathbf{J}(\vec{x})$ is the Jacobian matrix, $\vec{e}_{rep}(\vec{x})$ is the repeatable joint error vector, $\vec{e}_{r\bar{e}p}(\vec{x})$ is the nonrepeatable joint error vector, and $\vec{f}_{rep}(\vec{x})$ and $\vec{f}_{r\bar{e}p}(\vec{x})$ are all remaining repeatable and nonrepeatable errors repectively.

In this section we investigate the magnitude of endpoint position errors due to joint positioning errors for a spherical wrist; that is, we are only concerned with the terms in Equation D.1 involving the Jacobian. Assuming rotary joint positioning

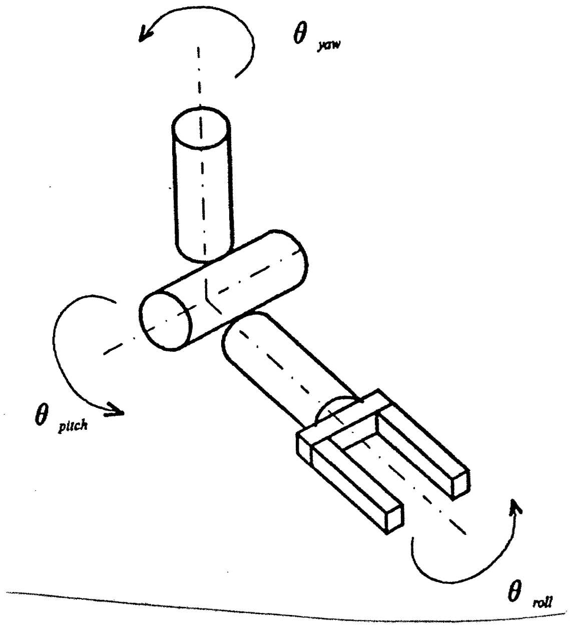


Figure D.1: Spherical Wrist Joint

errors in the pitch and yaw directions of 1.5 milliradians or $\frac{1}{4000}$ of a revolution,[1] the errors in positioning in the x, y, z and combined directions are shown in Figure D.2. Errors in the roll actuator do not produce position errors for parts centered in the grippers and are not included in the analysis. Note that the errors in the x and y directions produce complementary errors; that is, when combined they sum to a constant throughout the workspace.

If accuracy is more important in one direction rather than another, the best wrist configuration may be selected using the plots in Figure D.2. The lowest overall errors occur when the pitch axis is in the 0 position (grippers pointing straight down).

[1] This value was chosen because inexpensive commercial encoders are available to this accuracy. Higher accuracy encoders are also available.

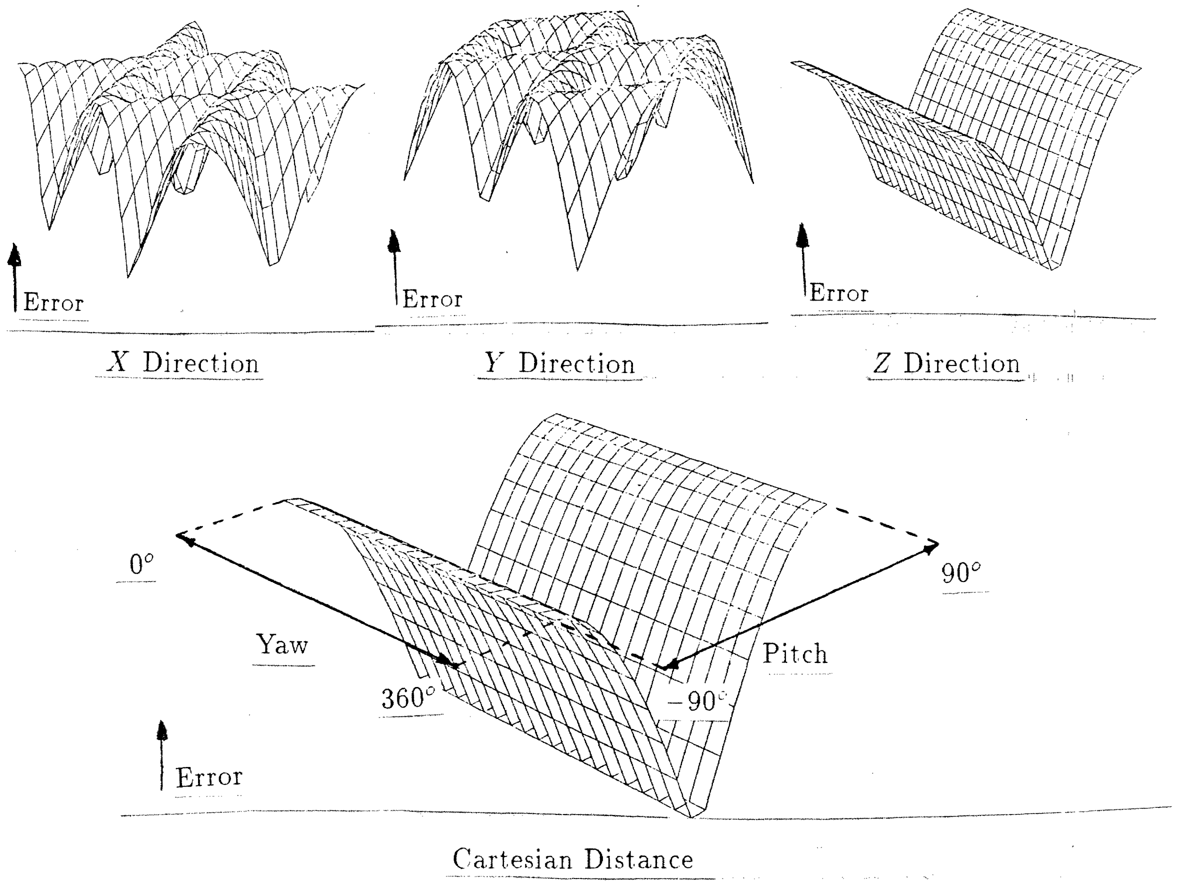


Figure D.2: Spherical wrist errors in the three coordinate directions and the cartesian sum. Errors are generated from yaw and pitch actuator errors of 1.5 milliradians.

Product of Two Normally Distributed Random Variables

Appendix E

In this section the probability density function for the product of two normally distributed random variables is derived. The sample space for the random variable

$$z = xy,$$

where x and y are also random variables, is shown in Figure E.1. The curve for a specific value of $z = z_0$ is shown, that is, the curve $x_0 = \frac{z_0}{y_0}$ is plotted, where x_0 and y_0 are specific values of random variables x and y respectively. The procedure given in [53] is followed for deriving the distribution $p_z(z_0)$ from the joint x and y distribution, $p_{x,y}(x_0, y_0)$ ($p_{x,y}(x_0, y_0) = p_x(x_0)\ p_y(y_0)$ for x and y linearly independent). First the cumulative density function, $p_{\leq z}(z_0)$, is found by integrating $p_{x,y}(x_0, y_0)$ over the portion of the sample space corresponding to $z \leq z_0$

$$p_{\leq z}(z_0) = \int_{y_0=0}^{\infty} \int_{x_0=-\infty}^{\frac{z_0}{y_0}} p_{x,y}(x_0, y_0)\ dx_0\ dy_0 + \int_{y_0=-\infty}^{0} \int_{x_0=\frac{z_0}{y_0}}^{\infty} p_{x,y}(x_0, y_0)\ dx_0\ dy_0. \tag{E.1}$$

Integrating over x_0, then differentiating with respect to z_0, we obtain an expression for the derived density function for z

$$p_z(z_0) = 2\int_{y_0=0}^{\infty} \frac{1}{y_0} p_{x,y}\left(\frac{z_0}{y_0}.\, y_0\right)\ dy_0. \tag{E.2}$$

We are interested in the case where x and y are independent normally distributed random variables with respective variances σ_x^2 and σ_y^2; thus, the joint density func-

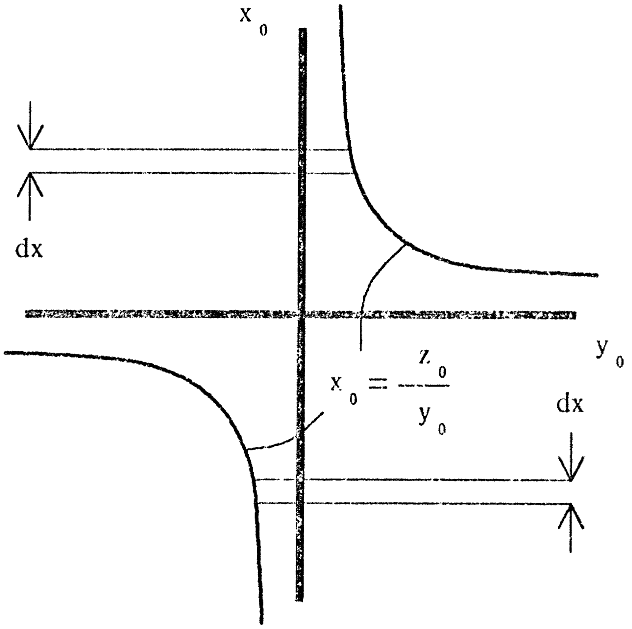


Figure E.1: Sample space for random variable $z = xy$.

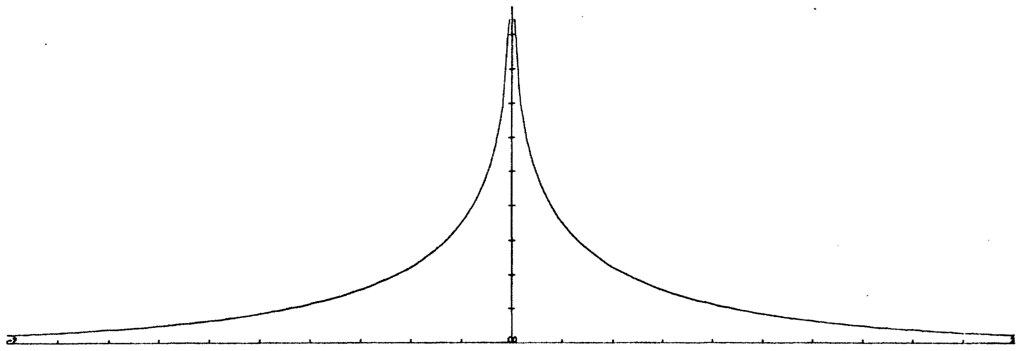


Figure E.2: Derived distribution for the product of two normally distributed independent random variables.

tion for x and y is

$$p_{x,y}(x_0, y_0) = \frac{1}{2\pi\sqrt{\sigma_x^2\sigma_y^2}} \exp\left(-\frac{1}{2}\left[\frac{x_0^2}{\sigma_x^2} + \frac{y_0^2}{\sigma_y^2}\right]\right).$$

Substituting this joint distribution into Equation (E.2)

$$p_z(z_0) = \frac{1}{\pi\sqrt{\sigma_x^2\sigma_y^2}} \int_{y_0=0}^{\infty} \frac{1}{y_0} \exp\left(-\frac{1}{2}\left[\frac{\left(\frac{z_0}{y_0}\right)^2}{\sigma_x^2} + \frac{y_0^2}{\sigma_y^2}\right]\right) dy_0. \qquad \text{(E.3)}$$

The solution to this equation may be found in a table of integrals [75]. The solution is

$$p_z(z_0) = \frac{1}{\pi\sqrt{\sigma_x^2\sigma_y^2}} K_0\left(\frac{|z_0|}{\sqrt{\sigma_x^2\sigma_y^2}}\right) \qquad \text{(E.4)}$$

where $K_0(x)$ is a modified Bessel function of the second kind, order 0 [87]. A plot of $p_z(z_0)$ is shown in Figure E.2.

Bibliography

[1] Abraham, R.G., "Programmable Automation of Batch Assembly Operations," *Industrial Robot*, v.4, no.3, Sept 1977.

[2] Acton, F.S., *Analysis of Straight Line Data*, Dover, New York, 1959.

[3] Agapakis, J.E., *Vision-aided Remote Robotic Welding*, Ph.D. Thesis, Dept. of Ocean Engineering, Mass. Inst. of Technology, 1985.

[4] Agin, G.J., "Calibration and Use of a Light Stripe Range Sensor Mounted on the Hand of a Robot," *IEEE Inter. Conf. on Robotics and Automation*, St. Louis, 1985.

[5] Agin, G.J., Binford T.O., "Computer Description of Curved Surfaces," *Proc. 3rd Int. Joint Conf. Artificial Intell.*, Stanford Univ. Aug 20-23, 1973, pp.629-640.

[6] Albano, A., "Representation of Digitized Contours in Terms of Conic Arc and Straight-Line Segments," *Computer Graphics and Image Processing*, v.3, 1974, pp.23-33.

[7] Albus, J., Kent, E., Nashman, M., Mansbach, P., Palumbo, L., "Six-dimensional Vision System", *SPIE Robot Vision*, v336, 1982.

[8] Allen, P., "Surface Description form Vision and Touch," *Proc. IEEE Int. Conf. on Robotics and Automation,* Atlanta, March, 1984.

[9] Allen, P., Bajcsy, R., "Two Sensors are Better than One: Example of Integration of Vision and Touch," *3rd Int. Symp. on Robotics Research*, 1985.

[10] Anderson, Robert H., "Programmable Automation: The Future of Computers in Manufacturing," Info. Sci. Inst. of USC, March 1973.

[11] Andreasen, M.M., "Design for Assembly - An Integrated Approach", *Assembly Automation*, v.2, no.3, Aug 1982.

[12] Arai, T., Endoh, T, Minokoshi, S., "Position and Orientation Measurements of a Moving Object by CCD Photo Array Sensors," *Proc. 4th Int. Conf. on Assembly Automation*, 1983.

[13] Asada, H., By, A.B., "Kinematic Analysis and Design for Automatic Workpart Fixturing in Flexible Assembly", *2nd Int. Symp. for Robotics Research*, Kyoto, Japan, 1984.

[14] Asada, H., Fields A., "Design of Flexible Fixtures Reconfigured by Robot Manipulators", *Proc. ASME Winter Annual Meeting*, Miami Beach, Nov. 1985, pp.251-257.

[15] Bajcsy, R., Allen, P., "Converging Disparate sensory Data," *2nd Int. Symp. on Robotics Research*, Kyoto, Japan, 1984.

[16] Ball, R.S., *A Treatise on the Theory of Screws*, Cambridge University Press, 1900.

[17] Ballard, D.H., Brown C.M., *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[18] Ballard, D.H., Sabbah, D., "On Shapes," *Proc. 7th Int. Joint Conf. Artificial Intell.*, 1981, pp.607-612.

[19] Baumeister, T., Avallone, E.A., Baumeister, T.,III (eds), *Marks' Standard Handbook rof Mechanical Engineers*, McGraw-Hill, New York, 1978.

[20] Beers, Y.B., *Introduction to the Theory of Error*, Addison-Wesley, Reading, Mass., 1957.

[21] Benedetti, M. "The Economics of Robots in Industrial Applications," *Industrial Robot*, v.4, no.3, Sept 1977.

[22] Beni, G., Hackwood, S., Trimmer, W.S., "High-Precision Robot System for Inspection and Testing of Electronic Devices," *Proc. IEEE Int. Conf. on Robotics and Automation,* Atlanta, March, 1984.

[23] Benton, R., Waters, D., *Intelligent Task Automation Interim Technical Reports,* AFWAL/MLTC Wright Patterson AFB, Ohio, Reports 1-7, Oct., 1985.

[24] Binford, T.O., "Survey of Model-Based Image Analysis Systems," *Int. J. of Robotics Research*, v.1, no.1, 1982.

[25] Bolle, R.M., Cooper, D.B., "On Optimally Combining Pieces of Information, with Application to Estimating 3-D Complex-Object Position from Range Data," *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8, no.5, 1986, pp.619-638.

[26] Bolles, R.C., Fischler, M.A., "A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data," *Proc. 7th Int. Joint Conf. Artificial Intell.*, 1981, pp.637-643.

[27] Bolles, R.C., Horaud, P., "Configuration Understanding in Range Data," *2nd Int. Symp. on Robotics Research*, Kyoto, Japan, 1984.

[28] Bolles, R.C., Horaud, P., Hannah, M.J.,, "3DPO: A Three Dimensional Part Orientation System," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[29] Bonner, S., and K. Shin, "A Comparative Study of Robot Languages", *IEEE Computer*, December, 1982.

[30] Bookstein, F.L., "Fitting Conic Sections to Scattered Data," *Computer Graphics and Image Processing*, v.9, 1979, pp.56-71.

[31] Boothroyd, G., *Design for Assembly - A Designers Handbook*, Dept. Mech. Eng., Univ. Mass, 1980.

[32] Boothroyd, G., Poli, C., Murch, L.E., *Feeding and Orienting Techniques for Small Parts*, Dep Mech Eng Univ Mass, 1980.

[33] Boothroyd, G., Poli, C., Murch, L.E., *Automatic Assembly*, Marcel Dekker, New York, 1982.

[34] Brooks, R.A., "Symbolic Error Analysis and Robot Planning", *Int. J. of Robotics Research*, v.1, no.4, 1982.

[35] Brou, P., "Using the Gaussian Image to find the Orientation of Objects," *Int. J. Robotics Res.*, v.3, no.4, 1984, pp.89-125.

[36] Buckley, S.J., *Planning and Teaching Compliant Motions,* Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, Mass. Inst. of Technology, 1987.

[37] Cain, R.A., "Inspection Using Planar Light," *Machine Intelligence Research Applied to Industrial Automation (Tenth Report),* SRI Int., Menlo Park, Calif., 1980, pp.33-187.

[38] Caine, M.E., "Chamferless Assembly of Rectangular Parts in Two and Three Dimensions," Masters Thesis, Dept. of Mechanical Engineering, Mass. Inst. of Technology, June, 1985.

[39] Camera, A., Migliardi,G.F., "Integrating Parts Inspection and Functional Control During Automatic Assembly," *Assembly Automation*, v.1, no.2, Feb 1981.

[40] Canny, J.F., "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8, no.6, 1986, pp.679-698.

[41] Canny, J.F., *Finding Edges and Lines in Images*, M.I.T. Artificial Intelligence Laboratory, Cambridge, Mass., Technical Report no.720, June 1983.

[42] Catalano, P. "Yes, There is a Labor Shortage," *Fusion*, May 1982.

[43] Charles Stark Draper Laboratory, *Proc. 3^{rd} Seminar on Assembly Automation,* Nov., 1982.

[44] Chin R.T., "Machine Vision for Discrete Part Handling in Industry; A Survey," *Proc. IEEE Workshop on Indust. Applications of Machine Vision,* May, 1982, pp.26-32.

[45] Clemens, D.T., *The Recognition of Two-Dimensional Modeled Objects in Images*, Masters Thesis, Dept. of Mechanical Engineering, Mass. Inst. of Technology, June, 1986.

[46] *Computerized Manufacturing Automation: Employment, Education, and the Workplace*, Washington, D.C.: U.S. Congress, Office of Technology Assessment, OTA-CIT-235, April 1984.

[47] Crandall, S.H., Karnopp, D.C., Kurtz, E.F., Jr., Pridmore-Brown, D.C., *Dynamics of Mechanical and Electromechanical Systems,* McGraw-Hill, New York, 1968.

[48] Datacube, Inc., Product literature, 4 Dearborn Road, Peabody, Mass. 01960.

[49] Davis, L.S., "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing*, v.4, 1975, pp.248-270.

[50] Dewhurst, P., Boothroyd, G., "Computer-Aided Design for Assembly," *Assembly Engineering*, v.26, no.2, Feb 1983.

[51] Djupmark, L. "Automation Research and Development Activities in Sweeden," *Assembly Automation*, v.2, no.2, May 1982.

[52] Dombre, E., Borrel, P., Liegeois, A., "A CAD System for Programming and Simulating Robots' Actions", *Digital Systems for Industrial Automation*, v.2, no.2, 1984.

[53] Drake, A.W., *Fundamentals of Applied Probability Theory*, McGraw-Hill, New York, 1967.

[54] Duda, R.O., Hart, P.E., "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. of ACM*, v.15, no.1, Jan., 1972.

[55] Durrant-Whyte, H.F., "Consistant Integration and Propagation of Disparate Sensor Observations," *IEEE Inter. Conf. on Robotics and Automation*, San Francisco, 1986, pp.1464-1469.

[56] Echigo, T., Masahiko, Y., "A Fast Method for Extraction of 3-D Information Using Multiple Stripes and Two Cameras," *Proc. 9th Int. Joint Conf. Artificial Intell.*, 1980,

[57] Elbracht, D., Schaler, H., "Automatic or Manual Assembly - Economic Considerations," *Assembly Automation*, v.2, no.3, Aug 1982.

[58] Faugeras, O.D., "New Steps Toward a Flexible 3-D Vision System for Robotics," *2nd Int. Symp. on Robotics Research*, Kyoto, Japan, 1984.

[59] Faugeras, O.D., Hébert, M., "A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces," *Proc. 8th Int. Joint Conf. Artificial Intell.*, 1983, pp.996-1002.

[60] Faugeras, O.D., Hébert, M., Pauchon, E., Ponce, J., "Object Representation, Identification, and Positioning from Range Data," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[61] Faux, I.D., Pratt, M.J., *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd., Chinchester, England, 1979.

[62] Fisher, E.L., Nof S.Y., Seidmann, "Robot System Analysis: Basic Concepts and Survey of Methods," source unknown.

[63] Gaston, P.C., Lozano-Perez T., "Tactile Recognition and Localization Using Object Models: The Case of Polyhedra in a Plane," *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-6, no.3, 1984, pp.257-265.

[64] Gelb, A. (ed.), *Applied Optimal Estimation*, MIT Press, Cambridge, Mass., 1974.

[65] Gershwin, S.B., "Material and Information Flow in an Advanced Automated Manufacturing System," MIT Lab. for Info. and Decis. Syst., LIDS-P-1199, 1982.

[66] Gevarter, W.B., "An Overview of Artificial Intelligence and Robotics Volume II - Robotics," Nat. Bureau of Standards, NBSIR 82-2479, 1982.

[67] Goad, C., "Robot and Vision Programming in ROBOCAM," *Proc. ASME Winter Annual Meeting*, Miami Beach, Nov. 1985, pp.219-224.

[68] Gordon, S.J., Seering, W.P., "Programmable Tools for Flexible Assembly Systems," *Proc. ASME Computers in Engineering*, Boston, Aug, 1985.

[69] Gordon, S.J., Seering, W.P., "Accuracy Issues in Measuring Quantized Images of Straight-Line Features," *IEEE Inter. Conf. on Robotics and Automation*, San Francisco, 1986, pp.931-936.

[70] Gordon, S.J., Seering, W.P., "Real-Time Part Position Sensing," submitted to *IEEE Trans. on Pattern Analysis and Machine Intelligence*, on Dec. 15, 1986.

[71] Gordon, S.J., Seering, W.P., "Locating Polyhedral Features from Sparse Light-Stripe Data," *IEEE Inter. Conf. on Robotics and Automation*, Raleigh N. Carolina, 1987.

[72] Gordon, S.J., Seering, W.P., "Position-Sensor-Driven Assembly," *IEEE Inter. Conf. on Robotics and Automation*, Raleigh N. Carolina, 1987.

[73] Gordon, S.J., Seering, W.P., Podoloff, R.M., "Programmable Automatic Assembly", *1^{st} Annual Report (prepared for IBM)*, M.I.T. Dept. of Mechanical Engineering, Nov. 1983.

[74] Gordon, S.J. (narrator), Tutsumi, S. (producer), "Prototype Flexible Assembly Cell", Video tape, M.I.T. Industrial Liason Program, Cambridge, Mass., 1985.

[75] Gradshteyn, I.S., Ryzhik, I.M., *Tables of Integrals, Series, and Products*, Academic Press, New York, 1980, p.342, eqn.3.478.4.

[76] Grimson, W.E.L., "Disambiguating Sensory Interpretations Using Minimal Sets of Sensory Data," *IEEE Inter. Conf. on Robotics and Automation*, San Francisco, 1986, pp.286-292.

[77] Grimson, W.E.L., Lozano-Perez, T., "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *Int. J. Robotics Res.*, v.3, no.3, 1984, pp.3-35.

[78] Grimson, W.E.L., Lozano-Perez, T., "Search and Sensing Strategies for Recognition and Location of Two and Three Dimensional Objects," *3^{nd} Int. Symp. on Robotics Research*, 1985.

[79] Grimson, W.E.L., Lozano-Perez, T., "Recognition and Localization of Overlapping Parts in Two and Three Dimensions," *IEEE Conf. on Robotics and Automation*, St. Louis, 1985, pp.61-66.

[80] Gustavson, R.E., "A theory for the Three-Dimensional Mating of Chamfered Cylindrical Parts," *J. Mechanisms, Transmissions, and Automated Design*, December, 1984.

[81] Hakala, D.G., Hilliard, R.C., Malraison, P.F., Nource, B.F., "Natural Quadrics in Mechanical Design," in *SIGGRAPH/81*, Dallas, Texas, 1981.

[82] Hébert, M., Kanade, T., "The 3-D Profile Method for Object Recognition," Internal paper Carnegie-Mellon University, 1984.

[83] Heginbotham, W.B. (ed.), "U.S. Unions - An Enlightened View of Robotics," *Industrial Robot*, v.7, no.4, Dec. 1980.

[84] Heginbotham, W.B. (ed.), "A Springboard for Future Flexible Assembly Systems," *Assembly Automation*, v.1, no.5, Nov 1981.

[85] Heginbotham, W.B, et al., "Flexible Assembly Module with Vision-Controlled Robot," *Assembly Automation*, v.2, no.1, Feb 1982.

[86] Herman, M., "Generating Detailed Scene Descriptions from Range Images," *IEEE Conf. on Robotics and Automation*, St. Louis, 1985, pp.426-431.

[87] Hildebrand, F.B., *Advanced Calculus for Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[88] Hill, J. "Dimensional Measurements from Quantized Images," *Machine Intelligence Research Applied to Industrial Automation (Tenth Report),* SRI Int., Menlo Park, Calif., 1980, pp.75-106.

[89] Hill, J.W., Sword, A.J., "Programmable Part Presenter Based on Computer Vision and Controlled Tumbling," *10th Int Symp Ind Rob.*

[90] Holland, S.W., Rossol, L., Ward, M.R., "CONSIGHT-I A Vision-controlled Robot System for Transferring Parts from Belt Conveyors," *Computer Vision and Sensor Based Robots*, (Dodd, G.G., Rossol, L. eds.), Plenum Press, New York, 1979.

[91] Horaud, P., Bolles, R.C., "3DPO's Strategy for Matching Three-Dimensional Objects in Range Data," *Proc. IEEE Int. Conf. on Robotics and Automation,* Atlanta, March, 1984.

[92] Horn, B.K.P., "The Binford-Horn Line-Finder", AI Memo 285, M.I.T. Artificial Intelligence Laboratory, Cambridge, Mass., 1971.

[93] Horn, B.K.P., *Robot Vision*, MIT Press, Cambridge, Mass. 1986.

[94] Hough, P.V.C., "Method and Means for Recognizing Complex Patterns," U.S. Patent 3,069,654, Dec. 18, 1962.

[95] Ikeuchi, K., Horn, B.K.P., Nagata, S., Callahan, T., Feingold, O., "Picking Up an Object from a Pile of Objects," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[96] Imaging Technology Inc., Product Literature, 600 west Cummings Park, Woburn, Mass. 01801. in Measuring Quantized Images of Straight Line Features," *IEEE Inter. Conf. on Robotics and Automation*, San Francisco, 1986, pp.931-936.

[97] Inoue, H, Inaba, M., "Hand Eye Coordination in Rope Handling," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[98] International Business Machines Incorporated, *A Manufacturing Language - Concepts and Users Guide*, December 1982.

[99] Jarvis, R.A., "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-5, no.2, March, 1983.

[100] Kanade, T., Sommer, T., "An Optical Proximity Sensor for Measuring Surface Position and Orientation for Robot Manipulation", *1st Int. Symp. on Robotics Research*, Bretton Woods, 1983.

[101] Kimemia, J.P., Gershwin S.B., MIT Lab. for Info. and Decis. Syst., LIDS-P-1134, 1982.

[102] Kinoshita, G., Idesawa, M., Naomi, S., "Robotic Range Sensor woth Projection of Bright Rong Pattern," *J. Robotic Systems*, v.3, no.3, 1986, pp.249-257.

[103] Klein, M.V., *Optics*, John Wiley and Sons, New York, 1970.

[104] Kondoleon, A.S., *Application of a Technological-Economic Model of Assembly Techniques to Programmable Assembly Machine Configuration*, MS Thesis, Dept. of Mechanical Engineering, Mass. Inst. of Technology, 1976.

[105] Korn,G.A., Korn,T.M., *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill, New York,1968.

[106] Koshikawa, K., A Polarimetric Approach to Shape Understanding of Glossy Objects," *Proc. 6th Int. Joint Conf. Artificial Intell.*, 1979, pp.493-495.

[107] Latombe, J.C., "Automatic Synthesis of Robot Programs from CAD Specifications", *Proc. NATO Advanced Study Institute on Robotics and Artificial Intelligence*, El Coccio, Italy, June 1983.

[108] Laugier, C., " A Program for Automatic Grasping of objects with a Robot Arm," *Proc. 11th Int. Symp. on Industrial Robots*, Tokyo, Oct. 1981.

[109] Lewis, J.W., "Product Design for Automatic Assembly," *Proc. Nat Des Conf*, 1982.

[110] Lieberman, L.I., Wesley, M.A., "AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly", *IBM J. of Research and Development*, v.21, no.4, 1977.

[111] Lin, X., Wee, W.G., "Shape Detection Using Range Data," *IEEE Conf. on Robotics and Automation*, St. Louis, 1985, pp.34-39.

[112] Lotter, B. "The Economic Design of Assembly Process," *Assembly Automation*, v.2, no.3, Aug 1982.

[113] Lozano-Perez, T., "Task Planning," in *Robot Motion: Planning and Control*, (Brady, M., et.al. eds.), MIT Press, Cambridge, Mass., 1982.

[114] Lozano-Perez, T., "Robot Programming", *Proc. IEEE Int. Conf. on Robotics and Automation*, 1983.

[115] Lozano-Perez, T., Mason, M.T., Taylor, R.H., "Automatic Synthesis of Fine Motion Strategies for Robots," AI Memo 759, M.I.T. Artificial Intelligence Laboratory, Cambridge, Mass., 1979.

[116] Lozano-Perez, T., Winston, P.H., "LAMA: A Language for Automatic Mechanical Assembly," *Proc. 5th Int. Joint Conf. Artificial Intelligence*, Mass. Inst. of Technology, Cambridge, Mass, 1977.

[117] Lozinski, C.A., *Robot Calibration*, Masters of Science thesis, Dept. of Mechanical Engineering, Mass. Institute of Technology, June, 1984.

[118] MacVicar-Whelan, P.J., Binford, T.O., "Line Finding with Subpixel Precision," *Proc. Image Understanding Workshop*, Science Applications Inc., Report no.SAI-82-391-WA, April, 1981.

[119] Maklin, A.G. "Using Vision in Electric Motor Assembly," *Assembly Automation*, v.1, no.3, May 1981.

[120] Marr D.C., *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman, San Francisco, CA, 1982.

[121] Marr D.C., Hildreth, E. *Theory of Edge Detection* Proc. R. Soc. Lond., B207, 1980, pp.187-217.

[122] Mason, M.T., Salisbury, J.K., Jr., *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, Mass, 1986.

[123] McCormick, D. (ed.), "Making Points with Robot Assembly," *Des Eng*, v.53, no.7, July/Aug 1982.

[124] McPherson, W., "Robotics: History and Development," *Science*, March 1982.

[125] Mendel, M., Gordon, S.J., "Bayesian Calibration and Estimation for Position Sensors," forthcoming.

[126] Merchant, E., "Flexible Manufacturing Systems," Paper presented at MIT Lab Manuf Prod, Dec 1982.

[127] Muller, Y., Mohr, R., "Planes and Quadrics Detection Using Hough Transform," *Proc 7th ICPR*, Montreal, Canada, 1984.

[128] Mundy, J.L., "The Limits of Accuracy for Range Sensing," *Proc. Workshop on Intelligent Robots: Achievements and Issues*, SRI Int. Menlo Park, Calif., July, 1985.

[129] Nakagawa, Y., "Automatic Inspection of Solder Joints on Printed Circuit Boards," SPIE Robot Vision, v.336, 1982.

[130] Nakagawa, Y., Takanori, N., "The Structured Light Method for Inspection of Solder Joints and Assembly Robot Vision Systems," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[131] Nevins J.L., "World Wide Activities - Industrial Robots and Assembly," Paper presented at 3rd Annual Seminar on Advanced Assembly Automation, Nov 1982.

[132] Nevins, J.L., Whitney, D.E., "Assembly Research," *Automatica*, v.16, pp 595-613, and in *Industrial Robot*, Mar 1980.

[133] Nevins, J.L., Whitney, D.E., "Computer Controlled Assembly", *Scientific American*, v.238, no.2, Feb 1978.

[134] Nishihara, H.K., Larson, N.G., "Towards a Real Time Implementation of the Marr and Poggio Stereo Matcher," *Proc. Image Understanding Workshop*, Science Applications, Inc., Report no. SAI-82-392-WA, April, 1981, pp.114-120.

[135] Nishihara, H.K., Pogio, T. "Stereo Vision for Robotics", *1st Int. Symp. on Robotics Research*, Bretton Woods, 1983.

[136] Nitzan, D., et. al., *Machine Intelligence Research Applied to Industrial Automation,* Technical Report no.12, SRI Project 2996, SRI Int., Menlo Park, Calif., 1983.

[137] Nof, S.Y., Fisher, E.L., "Analysis of Robot Work Characteristics," *Industrial Robot*, Sept 1982.

[138] Nof, S.Y., Knight, J.L., Salvendy, G., "Effective Utilization of Robot - A Job and Skills Analysis Approach," *AIIE Trans*, v.12, no.3, Sept 1980.

[139] Nof, S.Y., Lechtman, "Now Its Time For Rate-Fixing for Robots," *Industrial Robot*, June 1982.

[140] Nof, S.Y., Lechtman, "Robot Time and Motion System Provides Means of Evaluating Alternate Robot Work Methods," *Ind Eng*, Apr 1982.

[141] Nof, S.Y., Paul, R.L., "A Method of Advanced Planning of Assembly by Robots," *Proc Auto-Fact West*, Aneheim CA, Nov 1980.

[142] Nguyen, V.D., *The Synthesis of Stable Force-Closure Grasps*, Technical Report 905, M.I.T. Artificial Intelligence Laboratory, Cambridge, Mass, 1986.

[143] Ohwovoriole, M.S., "An Extension of Screw Theory and its Application to the Automation of Industrial Assemblies," Stanford Artificial Intelligence Lab Memo AIM-338, Apr., 1980.

[144] Oppenheim, A.V., Shafer, R.W., *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[145] Oshima, M., Shirai, Y., "Object Recognition Using Three-Dimensional Information," *Proc. 7th Int. Joint Conf. Artificial Intell.*, 1981, pp.601-606.

[146] Parthasarathy, S. Birk, J., Dessimoz, J., "Laser Range Finder for Robot Control and Inspection", *SPIE Robot Vision*, v.363, 1982.

[147] Paul, R.P., *Robot Manipulators: Mathematics, Programming, and Control,* MIT Press, Cambridge, Mass, 1981.

[148] Paul, R.P., Luh, J.Y., Nof, S.Y., "Advanced Industrial Robot Control Systems," Purdue University.

[149] Paul, R.P., Nof, S.Y., "Work Methods Measurement - A Comparison Between Robot and Human Task Performance," *Int J Prod Res*, v.17, no.3, 1979.

[150] Pavlidis, T., *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD., 1982.

[151] Pervin, E., Webb, J.A., "Quaternions in Computer Vision and Robotics," Department of Computer Science, Carnegie-Mellon Univ., CMU-CS-82-150, 1982.

[152] Pickett, M.S., Tilove, R.B., Shapiro, V., "ROBOTEACH: An Off-line Robot Programming System Based on GMSolid," *General Motors Research Publication GMR-4465*, 1983.

[153] Podoloff, R.M., Seering, W.P., "Design Considerations for a Robot End Effector for Automatic Assembly," *Control of Manufacturing Processed and Robotic Systems*, ASME, 1983.

[154] Popplestone, R.J., Brown, C.M., Ambler, A.P., Crawford, G.F., "Forming Models of Plane and Cylinder Faceted Bodies from Light Stripes," *Proc. 5th Int. Joint Conf. Artificial Intell.*, 1975, pp.664-668.

[155] Porter, G.B., Mundy, J.L., "Noncontact Profile Sensing System for Visual Inspection," *SPIE Robot Vision*, v.336, 1982.

[156] Porter, G.B., Mundy, J.L., "A Model Driven Visual Inspection Module," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[157] Ramer, U., "An Iterative Procedure for the Polygonal Appreoximation of Plane Curves," *Computer Graphics and Image Processing*, v.1, 1972 pp.244-256.

[158] Ranky, P.G., "Increasing Productivity with Robots in Flexible Manufacturing Systems," *Industrial Robot*, v.8, no.4, Dec 1981.

[159] Redford, A.H., "Robotic Assembly", presented at Automatic Assembly and Design for Assembly Conference - Univ Mass, Nov 1982.

[160] Redford, A.H., Lo, E.K., Killeen, P., "Cost Analysis for Multi-Arm Robotic Assembly", University of Salford Report, Feb. 1982.

[161] Rogers, P.F. "A Time and Motion Method for Industrial Robots," *Industrial Robot*, v.5, no.4, Dec 1978.

[162] Röcker, R., Kiessling, A., "Methods for Analyzing Three Dimensional Scenes," *Proc. 4th Int. Joint Conf. Artificial Intell.*, 1973, pp.669-673.

[163] Salamin E., "Applicaiton of Quaternions to Computation with Rotations," unpublished, internal memo, Stanford Artificial Intelligence Lab, 1979.

[164] Salisbury, J.K., Jr., *Kinematic and Force Analysis of Articulated Hands*, PhD. Thesis, Dept. of Mechanical Engineering, Stanford University, May, 1982; also Stanford Department of Computer Science Report No.STAN-CS-82-921; also in Reference [122].

[165] Salisbury, J.K., Craig, J.J., "Articulated Hands: Force Control and Kinematic Issues," Int. J. of Robotics Research, v.1, no.1, 1982.

[166] Sampson, P.D., "Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm," *Computer Graphics and Image Processing*, v.18, 1982, pp.97-108.

[167] Schröder, M., "Meeting a Variety of Future Needs with Flexible Assembly," *Assembly Automation*, v.2, no.1, Feb 1982.

[168] Seering, W.P., "Who Said Robots Should Work Like People?" *Technology Review*, April, 1985.

[169] Senker, P. "Social Implications of Automation," *Industrial Robot*, v.6, no.2, June 1979.

[170] Shaiken, H., "Technological Change and Shop Floor Response," unpublished case study.

[171] Shekhar S., Khatib, O., Shimojo M., "Sensor Fusion and Object Localization," *IEEE Inter. Conf. on Robotics and Automation*, San Francisco, 1986, pp.1623-1628.

[172] Shirai, Y., Inoue, H., "Guiding a Robot by Visual Feedback in Assembly Tasks," *Pattern Recog.*, v.5, 1973, pp.99-108.

[173] Shirai, Y., Koshikawa, K, Oshima, M., Ikeuchi, K., "An Approach to Object Recognition Using 3-D Models", *1st Int. Symp. on Robotics Research*, Bretton Woods, 1983.

[174] Shirai, Y., Koshikawa, K., Oshima, M., Ikeuchi, K., "An Approach to Object Recognition Using 3-D Solid Models," *Robotics Research: The First Symposium* (Brady, M. Paul, R. eds.), MIT Press, Cambridge, Mass. 1984.

[175] Shirai, Y., Suwa, M., "Recognition of Polyhedrons with a Range Finder," *Proc. 2nd Int. Joint Conf. Artificial Intell.*, 1971, pp.80-87.

[176] Simunovic, S.N., "An Information Approach to Parts Mating," PhD. Thesis, Dept. of Mechanical Engineering, Mass. Inst. of Technology, April, 1979.

[177] Smith, D., "Using Enhanced Spherical Images," AI Memo 451, M.I.T. Artificial Intelligence Laboratory, Cambridge, Mass., 1979.

[178] Smith, F.G., Thomson, J.H., *Optics*, John Wiley and Sons, Chinchester, 1971.

[179] Smith, R., Cheeseman, P., "On Representation and Estimation of Spatial Incertainty," *J. Robotics Research*, v.5, no.4, Dec., 1986.

[180] Smith, R., Self, M., Cheeseman, P., "Estimating Uncertain Spacial Relationships in Robotics," *IEEE Inter. Conf. on Robotics and Automation*, Raleigh, N. Carolina, 1987.

[181] Stauffer, R.N. (ed.), "Westinghouse Advances the Art of Assembly," *Robotics Today*, v.5, no.1, Feb 1983.

[182] Stone, H.W., Snaderson, A.C., Neuman, C.P., "Arm Signature Identification," *IEEE Inter. Conf. on Robotics and Automation*, San Francisco, 1986, pp.41-48.

[183] Sugihara, K., "Range Data Guided by a Junction Dictionary," *Artificial Intelligence*, v.12, 1979, pp.41-69.

[184] Sugihara, K., Okazaki, K., Kaihau, F., Sugie, N. "Regular Pattern Projection for Surface Measurement," *2nd Int. Symp. on Robotics Research*, Kyoto, Japan, 1984.

[185] Suzuki, T. Kohno, M., "The Flexible Parts feeder Which Helps a Robot Assemble Automatically," *Assembly Automation*, v.1, no.2, Feb 1981.

[186] Swansey, J.D., "Data-Driven Robotic Assembly, Retaining Accuracy Through Process and Software Techniques," *IEEE Conf. on Robotics and Automation*, St. Louis, 1985, pp.384-387.

[187] Takeyasu, K., Goto, T., Inoyama, T., "Precision Insertion Control Robot and Its Application," *ASME 76-DET-50*, 1976.

[188] Tanner, William R., "Basics of Robots," *Industrial Robot*, v.1, 1974.

[189] Taylor, R.H., "The Synthesis of Manipulator Control Programs from Task-level Specifications", Artificial Intelligence Laboratory, Stanford Univ., AIM 282, July 1976.

[190] Taylor, R.H., Hollis, R.L., Lavin, M.A., "Precise Manipulation with Endpoint Sensing," *IBM J. Res. Develop.*, v.29, no.4, July, 1985, pp.363-376.

[191] Taylor, R.H., Summers, P.D., Meyer, J.M., "AML: A Manufacturing Language", *Int. J. of Robotics Research*, v.1, no.3, 1982.

[192] Tesar, D., "Our Weakening Trade Position in Manufactured Goods: A Commentary on Mechanical Technology," *Prof Eng*, Aug 1979.

[193] Tomita, F., Kanade, T., "A 3-D Vision System: Generating and Matching Shape Descriptions in Range Images," *2nd Int. Symp. on Robotics Research*, Kyoto, Japan, 1984.

[194] Umetani, Y., "Research and Development of IR's in Japan," *Robots in the Japanese Economy*, Sadamato, K. (ed.) Tokyo: Survey Japan, 1981, pp 79-104.

[195] Vanderbrug, G.J., Albus, J.S., Barkmeyer, E., "A Vision System for Real Time Control of Robots," *Proc. 9th Int. Symp. Industrial Robots,* Washington, D.C., March, 1979.

[196] Wander, J., *The Application of Moire Interferometry to Automated Three Dimensional Inspection*, MS Thesis, Dept. of Mechanical Engineering, Mass. Inst. of Technology, Jan. 1985.

[197] Ward, A.C., *Robot Gripper Design*, Masters of Science thesis, Dept. of Mechanical Engineering, Mass. Institute of Technology, May 1984.

[198] Ward, M.R., Rossel, L., Holland, S.W., "Consight: A practical Vision-Based Robot Guidance System," *Proc. 9th Int. Symp. Industrial Robots,* Washington, D.C., March, 1979.

[199] Weiss, L.E. Sanderson, A.C., Neuman, C.P., "Dynamic Visual Servo Control of Robots: An Adaptive Image-Based Approach," *IEEE Conf. on Robotics and Automation*, St. Louis, 1985.

[200] Whitney, D.E., "Discrete Parts Assembly Automation – An Overview", *Proc. ASME Winter Annual Meeting*, Decembler 1978.

[201] Whitney, D.E., "Quasi-Static Assembly of Compliantly Supported Rigid Parts," in *Robot Motion: Planning and Control*, (Brady, M., et.al, eds.), MIT Press, Cambridge, Mass., 1982; also in *J. of Dynamic Systems, Measurement and Control*, v.104, March, 1982.

[202] Whitney, D.E., "Historical Perspective and State of the Art in Robot Force Control", *Proc. IEEE Conf. on Robotics and Automation*, St. Louis, 1985.

[203] Whitney, D.E., Lozinski, C.A., Rourke, J.W. "Industrial Robot Calibration Method," CSDL-P-1879, Charles Stark Draper Labs, Cambridge, Mass, 1984; also "Industrial Robot Calibration Method and Results," *Proc. ASME Computers in Engineering*, Las Vegas, 1984

[204] Whitney, D.E., Nevins, J.L., "What is the RCC and What Can It Do?", *9th Int. Symp. on Industrial Robots*, 1979.

[205] Will, P.M., Pennington, K.S., "Grid Coding: A processing Technique for Robot and Machine Vision", *Artificial Intelligence*, v.2, no.34, 1971, pp.319-129.

[206] Woodham, R.J., "Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from Image Intensity,", *Proc. 22nd Int. Symp. SPIE*, San Diego, Aug. 1978.

[207] Yonemoto, K., "The Art of Industrial Robots in Japan," *Proc of 11th Int Symp Industrial Robot*, Tokyo 1981, pp 1-7.

[208] Yonemoto, K., "The Socio-economic Impacts of Industrial Robots in Japan," *Industrial Robot*, vol 8, no.4, Dec 1981.

[209] Zimmerman, N.J., Scheerboom, P.L., Steenvoorden, C.K., Groen, F.C.A., "Automatic Visual Inspection System for Hybrid Circuits," *Proc. IEEE Workshop on Indust. Applications of Machine Vision,* May, 1982, pp.55-61.

125, 163, 164, 167, 169 - 172, 187,
192, 193, 197, 198, 204, 205, 214, 221,
224 - 226, 228 - 230, 248.

24 pages total
are missing.