SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION/DOWNGRADING SCHEDULE

3 DISTRIBUTION/AVAILABILITY OF REPORT
Unlimited Distribution

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
RSD-TR-29-86

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
College of Engineering
The University of Michiga

6b OFFICE SYMBOL
(If applicable)

7a NAME OF MONITORING ORGANIZATION
USAF/AFSC
AERONAUTICAL SYS DIV/PMRRC

6c ADDRESS (City, State and ZIP Code)
Ann Arbor, Michigan 48109

7b ADDRESS (City, State and ZIP Code)
WRIGHT-PATTERSON AFB, OH 45433-6503

8a NAME OF FUNDING/SPONSORING ORGANIZATION

8b OFFICE SYMBOL
(If applicable)

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F33615-85-C-5105

8c ADDRESS (City, State and ZIP Code)

10 SOURCE OF FUNDING NOS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT NO

11 TITLE (Include Security Classification)
Flexible Automatic Discrete Parts Assembly

12. PERSONAL AUTHOR(S)
D.E. Atkins, R.A. Volz

13a TYPE OF REPORT
Annual

13b TIME COVERED
FROM 8/1/85 TO 9/1/86

14 DATE OF REPORT (Yr., Mo., Day)
December 1986

15 PAGE COUNT
94

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB GR

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Manufacturing Science, Robotics, Artificial Intelligence, Sensors, Machine Vision, Special Purpose Processors, Motion Planning, Knowledge Based Systems, Requirements Analysis

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This project is focused on an important category of flexible automation problems, flexible automated assembly. The long term goal of this work is to develop technologies that can be applied to many different aspects of assembly automation and that can be evolved to successively higher levels of abstraction, i.e., that will eventually allow high level commands, such as "assemble product X", to be given. We are initially addressing three aspects of the problem, vision, planning/programming, and system integration. This report describes the first year's effort toward these goals.

- **A new method of smoothing surface surface data obtained from range scanners using Gaussian and other filtering methods has been developed. Initial test results are encouraging.**

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21 ABSTRACT SECURITY CLASSIFICATION

22a NAME OF RESPONSIBLE INDIVIDUAL

22b TELEPHONE NUMBER
(Include Area Code)

22c OFFICE SYMBOL

8. (continued) Robot Programming, Manufacturing Cell Architecture.

9. (continued)

- Techniques have been developed for estimating Gaussian curvature, mean curvature, critical points, and depth-discontinuity edges.
- Surface classification and region generation algorithms have been developed.
- We have shown that the Ego-motion Complex Log Mapping (ECLM) maintains projection invariance for arbitrary translational motion and can thus be used to recover depth information.
- Work on techniques for finding events in sequences of images has been successfully completed and reported.
- A computer controlled system for effectively achieving controlled camera motion has been developed so that knowledge based camera motion vision systems can be studied.
- An octree approach to collision free path planning for multiple robots has been formulated and an initial search technique developed.
- The graphic programming system was rewritten to make it more efficient and include an interactive editting and debugging capability using graphics simulation of program operation.
- The general problem solving methodology has been extended by developing: 1) a systematic approach for detecting the mutual subgoal constraints and ordering the subgoals in compliance with those constraints, 2) a procedure for automatically augmenting problem goal specifications to amke them more complete, and 3) a definition of a new heuristic estimation function into which the subgoal ordering constraints are incorporated.
- The assembly sequence problem for generalised screws has been formulated and a complexity analysis of the problem performed,
- A new approach has been introduced for solving fine motion problems in the face of uncertainties and control errors has been formulated in terms of the *type* of contact among the objects being assembled; the new approach reduces the dimensionality of the problem.
- The principle of developing *design constraints* whose satisfaction guarantees the success of sensor based verification and control methods has been introduced,
- An optimal part tolerancing problem has been formulated in a form that standard nonlinear programming methods may be used for its solution.
- Based upon a Semantic Association Model, a method has formulated for automatic determination of retrieving paths in an EDB/KLMS.
- A rapid prototyping "vision workbench" was developed.
- We have studied the mapping of several low- and mid- level vision algorithms onto hypercube parallel computers and performed benchmarks their performance on an NCUBE/6 machine, with encouraging results.
- We have identified the basic dimensions and issues involved in distributing the execution of Ada programs and proposed interpretations to complete the definition of the language with respect to distributed execution. A translator to allow distributed execution is in progress.

AUTHORS (continued)

PROFESSORS:

Keki B. Irani
Ramesh Jain
Arch W. Naylor

ASSOCIATE PROFESSORS:

Trevor N. Mudge
Kang Shin
Anthony C. Woo

ASSISTANT PROFESSORS:

Y. C. Lee
Terry Weymouth

RSD-TR-29-86

FLEXIBLE AUTOMATIC DISCRETE PARTS ASSEMBLY

ANNUAL REPORT

August 1, 1985 ~ September 1, 1986

Co-Principal Investigators

D. E. Atkins

R. A. Volz

Contract F33615-85-C-5105

EXECUTIVE SUMMARY
FLEXIBLE AUTOMATIC DISCRETE PARTS ASSEMBLY

Motivation and Goals

There is great emphasis today on the development of fully intergated manufacturing, including both the automation of individual manufacturing operations and the integration of all phases of manufacturing, from design through planning through production. The associated problems, however, are extremely complex and will require a large research effort over and extended period of time. This project is focused on an important category of flexible automation problems, in itself quite large, for which there are yet very few operational systems, flexible automated assembly. Assembly is the most highly labor intensive manufacturing process in the production of durable goods. There is thus great potential for direct cost saving through the development of flexible automated systems for assembly, as well as the indirect benefits mentioned above.

The long term goal of this work is to develop technologies that can be applied to many different aspects of assembly automation and that can be evolved to successively higher levels of abstraction, i.e., that will eventually allow high level commands, such as "assemble product X", to be given. We are initially addressing three aspects of the problem, vision, planning/programming, and system integration. This report describes the first year's effort toward these goals.

Context of Research

Figure .1 portrays a taxonomy of an automated assembly system into which our work fits. The boxes represent capabilities and operations that must be developed while inputs and outputs are shown in ovals. The vertical sequence of operations and outputs proceeding downward in the middle of the diagram illustrates the sequence of operations that must be performed in order to automatically assemble a product. The other boxes indicate essential capabilities which must be available in order to achieve the operation shown in the center. Few of the components shown are adequately developed at present. Moreover, many of the different constituents in an automatic assembly system will be implemented on different computers and, possibly, in different languages. Integration of these constituents into a working whole is an extremely important aspect of the overall problem.

The distinction between off-line and on-line activities is somewhat variable among

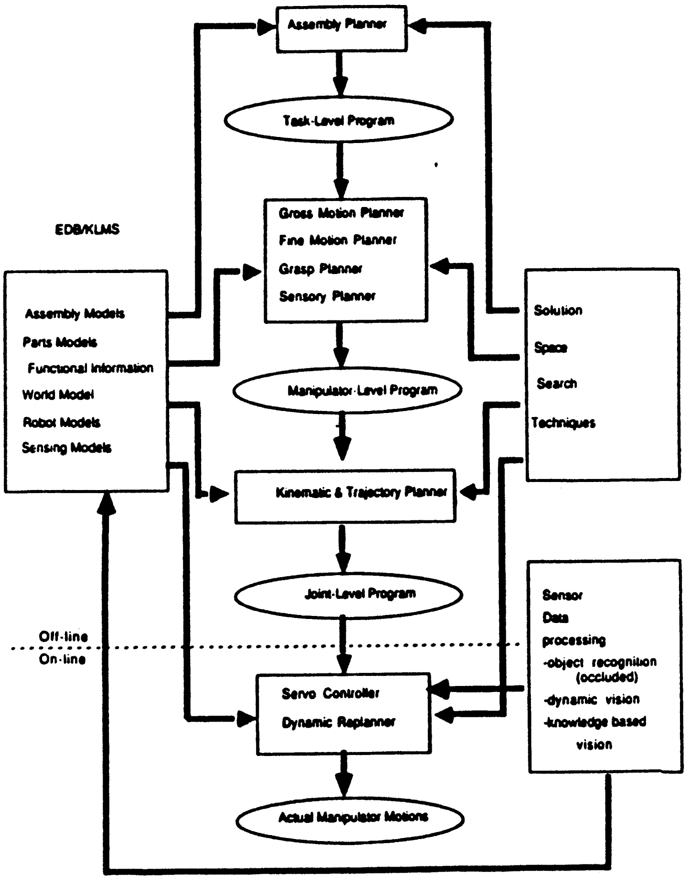


Figure .1: Interrelation between components in an automated assembly system

different systems, but as more sophisticated and complex planning algorithms are developed will tend to move downward toward the position shown in the figure. Our work addresses aspects of most of the off-line operations shown, the sensor data processing portion, primarily vision, of the on-line processing and the systems integration techniques by which such a system can be made operational. Our approach is to use existing technology, where available, coordinate with other research projects in the Robotics Research Laboratory and focus the research on this project toward key underlying technologies.

Highlights of Research Accomplishments

While the activities being reported cover the first year of this contract, they also, to a certain extent represent a transition from the previous contract. We thus report some activities which were completed during the year, work continuing from the previous contract, and new activities initiated during the first contract year. The following paragraphs briefly overview the principal accomplishments in each of the major research areas. It is important to recognise that in each of these areas there is significant additional work being done under the auspices of other contracts acquired, in part, due to the recognition and base of support provided by the Air Force.

Computer Vision

A flexible assembly system must be able to perceive its environment in order to take appropriate actions. Computer vision is being increasingly used to help machines achieve this perception. We are addressing several aspects of vision particularly important to manufacturing, including the following:

- Object recognition from partial views,
- Object recognition using range images,
- Dynamic scene analysis,
- Motion stereo,
- Qualitative vision, and
- Knowledge based vision techniques.

Specific achievements during this reporting period are:

- A new method of smoothing surface surface data obtained from range scanners using Gaussian and other filtering methods has been developed. Initial test results are encouraging.
- Techniques have been developed for estimating Gaussian curvature, mean curvature, critical points, and depth-discontinuity edges.
- Surface classification and region generation algorithms have been developed.
- We have shown that the Ego-motion Complex Log Mapping (ECLM) maintains projection invariance for arbitrary translational motion and can thus be used to recover depth information.
- Work on techniques for finding events in sequences of images has been successfully completed and reported.
- A computer controlled system for effectively achieving controlled camera motion has been developed so that knowledge based camera motion vision systems can be studied.

Planning and Programming

As is evident from Figure .1 there are several levels of planning and support tools that are necessary to achieve automated assembly. The areas investigated during the past year include:

- Gross motion path planning,
- Assembly sequence planning,
- Planning how to use sensor information to overcome uncertainties in geometries (tolerances), sensor errors, and control errors,
- Tolerancing,
- Graphical robot programming,
- Heuristic problem solving, and
- Development of an Engineering Database/Knowledge Management System.

Accomplishments during the past year include:

- An octree approach to collision free path planning for multiple robots has been formulated and an initial search technique developed.
- The graphic programming system was rewritten to make it more efficient and include an interactive editing and debugging capability using graphics simulation of program operation.
- The general problem solving methodology has been extended by developing: 1) a systematic approach for detecting the mutual subgoal constraints and ordering the subgoals in compliance with those constraints, 2) a procedure for automatically augmenting problem goal specifications to make them more complete, and 3) a definition of a new heuristic estimation function into which the subgoal ordering constraints are incorporated.
- The assembly sequence problem for generalised screws has been formulated and a complexity analysis of the problem performed,
- A new approach has been introduced for solving fine motion problems in the face of uncertainties and control errors has been formulated in terms of the *type* of contact among the objects being assembled; the new approach reduces the dimensionality of the problem.

- The principle of developing ***design constraints*** whose satisfaction guarantees the success of sensor based verification and control methods has been introduced,

- An optimal part tolerancing problem has been formulated in a form that standard nonlinear programming methods may be used for its solution.

- Based upon a Semantic Association Model, a method has formulated for automatic determination of retrieving paths in an EDB/KLMS.

Systems Integration and Architecture Techniques

Systems integration is essential for any robotics or manufacturing system, but is today done with little more than ad hoc tools, and often starting from scratch each time a system is built. Moreover, there is little experience with the use of newer forms of parallel architectures for robotics and manufacturing applications. In this area we are investigating:

- A robotics rapid prototyping system,

- The mapping of (vision) algorithms unto parallel architectures, and

- Distributed Language techniques, which we believe to be fundamental to developing a systematic way of accomplishing systems integration.

Specific accomplishments during the past year include:

- A rapid prototyping "vision workbench" was developed.

- We have studied the mapping of several low- and mid- level vision algorithms onto hypercube parallel computers and performed benchmarks their performance on an NCUBE/6 machine, with encouraging results.

- We have identified the basic dimensions and issues involved in distributing the execution of Ada programs and proposed interpretations to complete the definition of the language with respect to distributed execution. A translator to allow distributed execution is in progress.

Organisational Considerations

This year has seen a continued excellent performance in terms of student participation and technical publication. Summary statistics directly supported by Air Force funding are:

- 21 papers presented at major conferences,
- 14 papers appeared in review journals,
- 13 papers are currently under review or have been accepted for journal publication during the coming year,
- 8 papers are currently under review or have been accepted for conference presentation.
- 15 graduate students participated in the program,
- 5 students participating in the program received their Ph.D.'s,
- 1 student received the M.S. degree; the lower number here reflects the fact that the students receiving support from the program are more senior than in previous years.

TABLE OF CONTENTS

FLEXIBLE AUTOMATIC DISCRETE PARTS ASSEMBLY

1 Introduction

1.1 Motivation and Goals

There is great emphasis today on the development of fully intergated manufacturing. Indeed, the development of integrated manufacturing techniques is critical to the survival of many segments of United States' industry. Integrated manufacturing includes both the automation of individual manufacturing operations and the integration of all phases of manufacturing, from design through planning through production. The motivations for integrated manufacturing, ultimately, are economic, though there are numerous intermediate reasons such as improved quality, reduced lead time on introducing new products, reduced inventories (through the ability to quickly produce new parts when needed) and the ability to rapidly prototype new designs, in addition to the traditional reason of reduced direct production cost.

This project is focused on an important category of flexible automation problems for which there are very few operational systems, flexible automated assembly. Assembly is the most highly labor intensive manufacturing process in the production of durable goods [1]. There is thus great potential for direct cost saving through the development of flexible automated systems for assembly, as well as the indirect benefits mentioned above.

There are many levels of assembly automation one can seek. Ultimately one would like to be able to give a very high level command such as "Build 30 of product X by deadline," and have the system automatically schedule the assembly process, order the components and tools needed, schedule their delivery to the work area, determine the assembly sequence and fixturing required, generate the robot programs required, and manage the flow of all material and information to accomplish the assembly. The achievement of this level of automatic operation is a long way off. A slightly lower, but also very difficult level is the task level, in which one would like to be able to give commands such as "assemble product X," "insert bolt A in hole B," or "grasp bracket C" under the assumption that fixturing, materials, and tooling are all available inputs; even within this level there are obviously many sublevels.

The long term goal of our work is to develop technologies that can be applied to any of several levels of assembly automation and can be evolved to higher levels of abstraction. In particular we are addressing three aspects of the problem, vision, planning/programming, and system integration.

1.2 Overview of Automatic Assembly

Figure 1 portrays the major components which must be part of an automated assembly system at the task level (i.e., the "assembly product X" level), and the relationships between them. The boxes represent capabilities and operations while inputs and outputs are shown in the ovals. The vertical sequence of operations and outputs proceeding downward in the middle of the diagram illustrates the sequence of operation which must be performed in order to automatically assembly a product. The other boxes indicate essential capabilities which must be available in order to achieve the operation shown in the center. Few of the components shown are adequately developed at present. Indeed, even the appropriate contents of the database are not fully known at this time.

We assume as inputs to the process a wide variety of design, work cell model and sensor information. In order to manage this information, an Engineering Database/Knowledge Management System (EDB/KMS) is necessary that combines the data storage/retrieval capabilities normally present in database management system with semantics extraction and inferential mechanisms normally associated with knowledge management systems. Further, the EDB/KMS must support access at various levels of abstraction with corresponding levels of access times.

The assembly planner produces the sequence in which the components to the product are to be assembled, and the associated fixturing and orientation information for each step in the sequence, called the (a subtask level program) in Figure 1. Each step at the subask level consists of a single operation such as placing a part in a fixture, changing a tool on a robot end effector, or mating two parts.

Typically there is a robot motion associated with each subtask step resulting from the assembly sequence planner. Each such subtask requires several further stages of planning. First, the motion form the initial position of the robot to a position in free space near its final position must be determined which avoids collision and minimizes an appropriate cost function, e.g., time or energy (gross motion planning). Next motions involving contact among the parts being mated must be planned (fine motion planning). These two planning stages are separated because some gross motion planners [2] do not permit contact (though some some other cases the two might be performed jointly [3]. Also, if a component is to be picked up by a robot, the point at which it is to be grasped must be chosen so that no interference occurs among the hand, the component

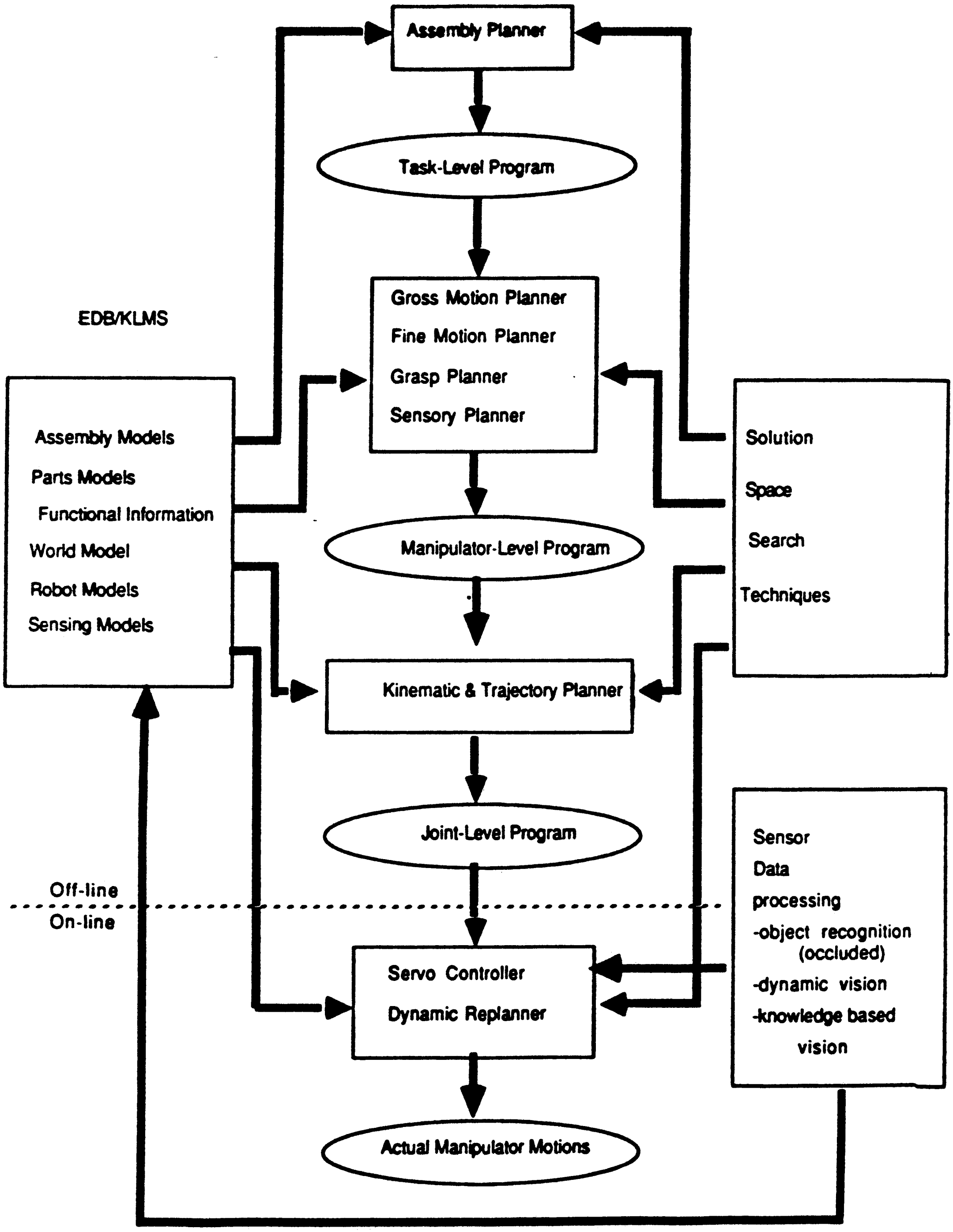

Assembly Planner
Task-Level Program
Gross Motion Planner
Fine Motion Planner
Grasp Planner
Sensory Planner
EDB/KLMS
Assembly Models
Parts Models
Functional Information
World Model
Robot Models
Sensing Models
Solution
Space
Search
Techniques
Manipulator-Level Program
Kinematic & Trajectory Planner
Joint-Level Program
Sensor
Data
processing
-object recognition
(occluded)
-dynamic vision
-knowledge based
vision
Off-line
On-line
Servo Controller
Dynamic Replanner
Actual Manipulator Motions

Figure 1:

and other parts of the assembly, and also so that the component will not slip in the grasp of the robot. Finally, at this stage of planning, sensor plans must be developed which can use the robot's sensors to determine whether or not a subtask step was successfully completed.

From the perspective of the robots, the output of subtask planning stage is a sequence of specific motion, gripper and sensor commands in a world coordinate system of the workspace. The kinematic and trajectory planner converts these motions into joint motions of the robots involved and determines the planned time history of each joint along the desired path.

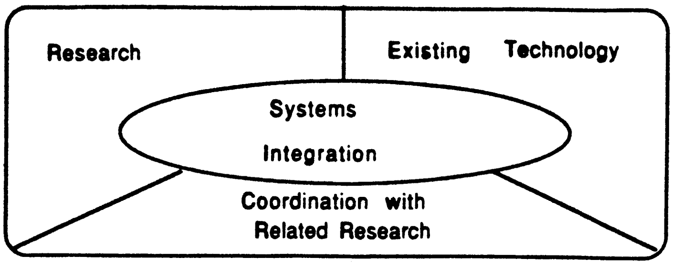
The servo level, of course, directs the movement of each joint of the robots. It will be important for assembly tasks that the control system be capable of compliant motion (e.g., move in a given direction while maintaining a prescribed contact force). Guarded motions, i.e., motions that continue until some sensed force, torque, or position condition is satisfied, will also be important and, indeed, are closely related to the sensor planning for subtask verification. Essentially, as each guard condition satisfied and a motion stopped, the sensor readings are compared with planned values to determine whether or not the desired motion actually occurred. Unfortunately, uncertainties in part geometries, actual gripping position, sensor errors, control errors, etc., fairly frequently result in a motion being stopped due to excessive force or torque. When this is detected, as a result of the sensor verification plan, a dynamic fine motion replanning is necessary. Since presumably the robot is close to the desired position, this planner can be much simpler, and hence done in real-time, than the original planner.

Several of the planning stages required searching extensive solution spaces. Efficient techniques are necessary if automatic planning is to be realistically achieved.

It is obvious that a multitude of sensing is required for successful operation of an assembly system. Vision, force/torques, tactile and range sensors are the primary candidates, though if the assembly involves are welding, temperature sensing (of the weld puddle) may prove useful. Many of the actual motion target positions will be determined from sensor information, some servoing of the motions may be driven by higher level sensors (e.g., vision) and certainly sensing is required for verification.

Finally, many of the different constituents in an automatic assembly system will be implemented on different computers and, possibly, in different languages. Integration of these constituents into a working whole is an extremely important aspect of the overall problem.

Relation of Research to Existing Activities



Air Force supported research
- Vision and sensing
- Programming and automatic planning
- Systems integration and rapid prototyping technology

Coordination with related research
- Development of distributed systems integration language
- Relation of (vision) algorithms and architectures
- Automatic robot program modification to accommodate component redesign
- Development of tactile sensor

Use of existing technologies
- robots, end effectors and controls
- sensors

Figure 2:

1.3 Project Organization and Activities

Fully automated discrete parts assembly is a very complex problem and will require a large research effort over an extended period of time. Our approach is to use existing technology, where available, coordinate with other research projects in the Robotics Research Laboratory and direct the research on this project toward key underlying technologies. This concept is illustrated in Figure 2. In particular, our research emphasis may be grouped in three important categories, machine vision and sensing, programming and planning, and systems integration techniques. Figure 3 shows the organization of the project in greater detail, and identifies the major subprojects.

The machine vision research is exploring a variety of techniques ultimately useful for object recognition and pose determination. This is being studied via single image analysis, the analysis of a sequence of images, and the use of knowledge based approaches and with both grey level and range images as inputs. Results obtained for occluded part recognition have been shown to outperform previously known techniques, the fundamental basis for differential geometric descriptions and segmentation of surfaces in range images has been established, and promising work has been started on

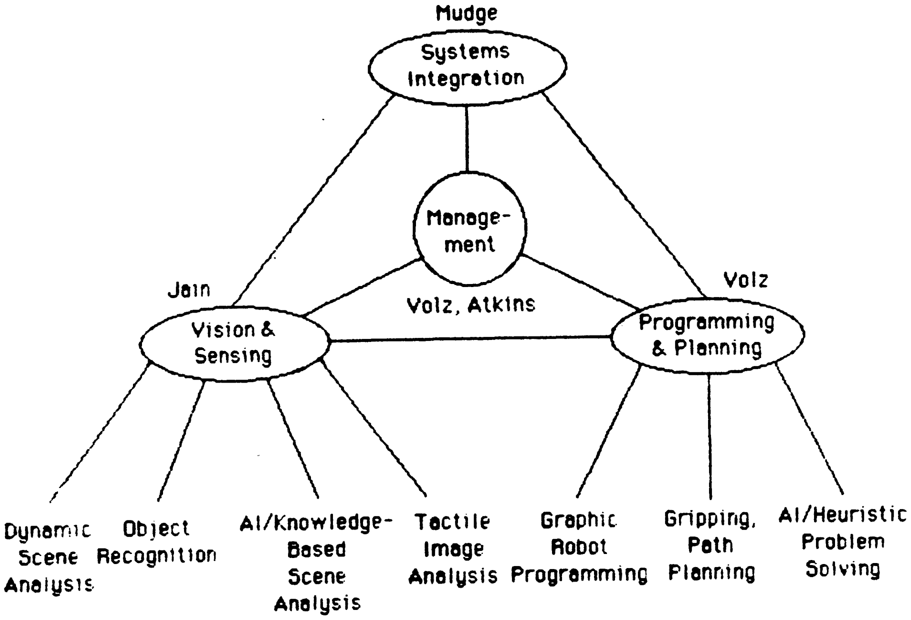


Figure 3:

the mapping of vision algorithms onto parallel processors of a hypercube architecture.

Work in the planning and programming area has produced significant results in path planning, grasping and heuristic problem solving techniques (which are continuations of previous work) and began important new work in assembly and sensor planning was begun. The extension of parallel finger grasping to include nonuniform grasping pressure, general loading forces and accurate computation of torques resisting slippage has been more fully tested and completed. In the area of general problem solving, the techniques obtained have been shown to be very robust, exhibiting performance equal to the best individual domain specific solution obtained across a broad range of problems.

To date there has been almost no work on automatic planning of assembly sequences. We are initially approaching the problem by considering a restricted class of assembly tasks involving primarily axis oriented operations, i.e., placing parts (e.g. brackets, washers, nuts) on a shaft of some type. In the area of sensor planning we are introducing two important new principals: 1) establishing relations between assembly design constraints and the use of sensors to guide the assembly operation in the present of uncertainties, and 2) the reduction of the problem dimensionality through the use of ***contact formations*** as the basis for verification and replanning rather than six dimensional configurations. This effort is only in its early stages, but initial results are encouraging and some design constraints which allow certain formations to be distinguished, even in the presence of uncertainties, have been established.

The systems integrations work has had two aspects: 1) building a rapid prototyping robot/vision system which can be used as the basis for future experimentation and research, and 2) serving as a focussing agent for systems integration techniques based around distributed languages (largely supported by other contracts). An initial version of the rapid prototyping system is operational. Image acquisition and edge detection essentially proceeds in real-time, making the complete real-time implementation of vision algorithms a real possibility in our future work. In view of the importance of and need for systems integration techniques for distributed systems and the role of this project as a focussing agent for our work in this area, our vision of distributed manufacturing software will be described and related activities described briefly.

References

[1] A.G. Boothroyd. "Use of Robots in Assembly Automation," *Annals of the CRP,* Vol. 3, No. 2, 1984.

[2] E.G. Gilbert and D.W. Johnson. "Distance Functions and their Application to Robot Path Planning," *IEEE Journal of Robotics and Automation,* Vol. RA-1, No. 1, pp. 21-30, March 1985.

[3] Tomas Lozano-Peres, and M.T. Mason and R.H. Taylor. "Automatic Synthesis of Fine Motion Strategies for Robots," *The Inter. Jour. of Robotics Research,* Vol. 3, No. 1, pp 3-24, 1984.

2 Research

2.1 Computer Vision

A flexible assembly system should be able to perceive the environment to take appropriate actions. Computer vision is being increasingly used to help machines perceive their environment. In our project, we are addressing several relevant aspects of computer vision. We are addressing the problem of object recognition from their partial views. We have started addressing the object recognition problem using range images. We continue to address problems in dynamic scene analysis and have made significant progress in our implementation of motion stereo and our understanding of qualitative vision. Finally, we are exploring application of knowledge based techniques to vision systems for assembly.

2.1.1 Range Image Understanding

Investigator: R. Jain

Research objective

In most applications, a 3-dimensional object must be recognized from its 2-dimensional projections. In a scene, there may be several objects which for some reason, may be only partially visible in an image obtained from a certain viewpoint. Conventional global features, such as areas, moments, Fourier descriptors, cannot be used to recognize objects from only partial views. Local features like corners and critical points have been proposed for recognizing objects. The success of such local features has been limited due to the sensitivity of methods to determine them and the difficulties in matching.

A surface descriptor should capture the intrinsic nature of the surface as a global feature. Such a descriptor will be very useful in object recognition if it can be computed from partial views of object surfaces. Commonly used local and global features in object recognition lack this desirable characteristic.

Status of Project

Depth maps are first smoothed using Gaussian and other filtering methods to remove noise present in the input data. This smoothed data is then interpreted as samples of an underlying surface. A problem in this simple smoothing arises at the edges of surfaces. Clearly, one should apply smoothing operations only to points belonging to the same surface. This is not easy because smoothing precedes segmentation and, therefore, the surfaces are not yet known. We implement surface-based smoothing in an indirect way in our point grouping operation by detecting a kernel of a surface and then growing the surface based on the raw data. This approach uses smoothing to obtain the kernel, but relies on the original data to obtain surfaces. Note that a surface is obtained by fitting the best surface to many points and, hence, noise is removed during that stage. We have already experimented with this approach with very encouraging results. We will rigorously study this new alternative surface based smoothing to understand its scope and limitations.

The derivative estimates are used to determine Gaussian curvature, mean curvature, critical points, and depth-discontinuity edges. Roof-edges are detected by high mean curvature regions. A critical points image is generated using the intersection of the zero-crossing images of the two first partial derivatives. We have studied these in detail and published an article that appeared in *Computer Vision, Graphics, and Image Processing*[1]. In this paper, we introduced the concept of s-curvature characteristics for representing differential geometric characteristics of surfaces. We demonstrated that these characteristics allow determination of different types of edges and may plan an important role in the study of surfaces.

The sign of the surface curvature values can be used to place every pixel in one of eight classes: pit, peak, saddle ridge, saddle valley, valley, ridge, minimal, and flat (planar). Our experience with this classification is extremely encouraging. In most cases, the classification appeared to be intuitively correct. We will study how the characterization of points into basic surface types is effected when data is taken from different range sensors and when noise is added to synthetic images.

The next step is grouping points into surfaces. In the preceding step a point was assigned to one of the eight basic surface types. This type and the spatial proximity of points are two strong factors for grouping points. We developed an approach for segmentation that combines features of region growing, random consensus, and surface fitting[2]. We start with a connected component comprised of points with the same basic surface type, and reduce this component to a kernel which contains points that have a high probability of belonging to the same surface according to some criterion.

This is accomplished using iterative erosion of the region until it is reduced to a small set of points that most likely comes from the same surface. Starting with these points, the region is grown using the best approximating surface. In region growing, we consider the points that are neighbors of the points that have already been grouped. If the error in surface fitting when these points are included in the group is within an acceptable bound, then the new points are included in the group. For surface approximation, we use planar, biquadratic, and biquartic polynomial surfaces. The algorithm selects the order of surface depending on the closeness of fit. It starts with a planar surface, and goes to the next higher order surface if required. Though we are using only three types of surfaces, more surfaces could be used without any problem. We feel, however, that surfaces beyond biquartic may not be required for object recognition. We should point out that this variable order surface fitting feature of the algorithm allows the 'best' rather than predetermined surface to be fitted to the data. This algorithm has been implemented and our experiments with several real images show its efficacy.

Future Research

Some of the problems related to this segmentation will be studied during the proposed period. First, there are several possible error criteria; we will study these to determine their performance. The order in which points are considered for grouping may also be evaluated based on their different surface characteristics. Finally, the termination criterion for the region growing will also pose an interesting problem. Besides conventional approaches for termination, such as the rate of growth, we will also consider surface characteristics for termination.

We expect to develop a robust methods to recover symbolic descriptors for surfaces present in an image, and to demonstrate the efficacy of these symbolic descriptors in object recognition. The next phase of the research will be concerned with obtaining descriptors from models and developing approaches for matching to recognized objects. This phase will not be addressed in the revised research plan.

References

[1] P. Besl and R. Jain. "Invariant surface characteristics for 3-D object recognition in depth maps," *Computer Vision, Graphics and Image Processing,* Vol. 33, pp. 33-80, 1986.

[2] P. Besl and R. Jain. "Segmentation through Symbolic Surface Descriptions," to appear in *IEEE Trans. on Pattern Analysis and Machine Intelligence.*

2.1.2 Dynamic Scene Analysis

Investigator: R. Jain

Research Objective

Stereo information can be obtained using a moving camera. If a dynamic scene is acquired using a translating camera and the camera motion parameters are known, then the analysis of the scene may be facilitated by Ego-Motion Complex Logarithmic Mapping (ECLM). We are developing an approach to motion stereo that will allow segmentation and depth recovery in the ECLM space.

Considering the sensitivity of the approaches for structure from motion, we have started exploring feasibility of *qualitative vision*. In qualitative vision, our aim is to see how approximate methods may be used in dynamic environments. We believe that over an extended time period, the information recovered using a qualitative approach will be more useful and reliable than noise-sensitive quantitative methods currently being studied by many researchers for structure from motion.

Status of Project

Depth determination is a continuing problem in computer vision. Research in this area is motivated by the perceived need for autonomous vehicles, vision for the blind, realistic flight trainers, models of the human visual system, etc. There is a plethora of depth determination techniques. Many different stereo systems for depth determination have been developed just in the last few years. Optical flow and structure from motion have attracted many researchers in dynamic scene analysis. Many efforts have been made to develop techniques that will allow recovery of depth information in sequences acquired using a moving camera. The optical flow based technique faces an unsolved problem of determining good quality optical flow for real scenes. The structure from motion techniques is too sensitive to noise.

Jain et al [1], [2] initiated some studies using the same moving camera configuration that some earlier researchers studied. In this configuration, the camera has only translational motion and the velocity component along the optical axis is nonsero. However, they approached this problem very differently; they map images into a complex log space where the movement of the objects in two dimensions due to the camera motion becomes a translation along one axis in the new space. Given this phenomenon, the correspondence problem is greatly reduced, since only a small strip of the new space needs to be searched. This constraint is similar to the epi-polar constraint in stereo. In addition, the transform is scale and rotation invariant. It also has an analogue in the human visual system – the mapping of the retinal space into the striate cortex is very closely approximated by the CLM. This mapping based approach allows us to use important characteristics of optical flow but we don't need to determine optical flow. This is because we are using constrained motion of the camera and utilising the known ego-motion. Thus, we are using known motion to bypass a difficult problem.

We have shown that the ECLM maintains projection invariance for arbitrary translational motion of the observer and can be used in recovering the depth of stationary objects[1]. We studied aspects of the mapping that are useful in finding the precision of the information that can be recovered using this mapping. We did several experiments in recovering depth in laboratory scenes[2]. These experiments are very encouraging. They demonstrate that this technique may play a very important role in many industrial environments. The results of our findings were presented in papers that are going to appear in books and journals.

Dynamic scene analysis may be facilitated by a long sequence. Using reliable qualitative information, such as occlusion and sense of motion, over an extended sequence, it is possible to build dynamic models of a scene. such models may not contain precise metric information, but will be very useful in focus of attention mechanism. Rigorous processes may, then, be applied to these parts of the scene where precise metric information is required. Motivated by this, we started exploring a qualitative approach for building dynamic scene models.

In qualitative vision, we finished our efforts to find events in a sequence using successive refinement of parameters. This work has been reported in an article [3]. Currently, we are implementing a non-monotonic temporal reasoning approach to use only the direction of motion and the occlusion information for building the world model giving relative depth information of the objects in a dynamic scene.

Future Research

Our future research efforts in qualitative vision will focus in implementing the reasoning system using laboratory sequences. This system will use information obtained from other vision programs. In motion stereo, our efforts will be directed towards refining depth recovery using multiple frames and combining segmentation and depth recovery in the ECLM space. We will use the dynamic models given by the qualitative approach to focus attention and to determine camera motion for recovering information. We will study the performance of motion stereo and apply it in assembly applications.

References

[1] R. Jain and N. O'Brien. "Ego-Motion Complex Log Mapping." *SPIE's Intelligent Robots and Computer Vision Conference,* November 1984.

[2] R. Jain, S. Bartlett and N. O'Brien. "Motion Stereo Using Ego-Motion Complex Logarithmic Mapping," accepted for publication in *IEEE Trans. on Pattern Analysis and Machine Intelligence.*

[3] S. M. Haynes and R. Jain. "Event Detection and Correspondence," *Optical Engineering,* Vol. 25, No. 3, pp. 387-393, 1986.

2.1.3 Knowledge Based Computer Vision

Investigator: T. Weymouth

Introduction

In the interpretation of dynamic scenes much uncertainty arises from errors in measurement, mistakes in processing, and misapplied assumptions. This uncertainty can be overcome by the flexible application of knowledge. A system which uses knowledge of object geometry, expected object placement and attitude, and the relations between object features and image measurements can effectively " fill in the gaps" in interpretation.

In using knowledge, one key problem is control. Machine vision deals with an active environment and there must be methods to make a timely selection of the relevant knowledge and processes for each part of the interpretation. The environment includes objects in a background, the sensory mechanism (the camera), and the processes that are available for interpretation. Any part of this environment may be under the direct control of the interpretation process. Objects and the camera can be moved under computer control, the camera parameters can be altered, processes can be selected, and the portion of sensory data to be used can be selected. Each of the options implies choices which require control.

We might better see the effect of this control by characterizing machine vision as consisting of five stages: data collection, data reduction, interpretation, planning and action. The relation between stages is complex, but can be summarised by stating that each stage precedes the following and that action leads to the collection of more data. In the data collection stage sensory information is gathered and put into some internal form (e.g. an image array). As this is usually much more data than the system can handle, some data reduction takes place over those extracted entities and elicits a plan of action. The resulting plan causes a change in the environment: objects move, the camera is repositioned or alternate processes are selected. These changes afford the opportunity for additional information. Thus, the cycle is started again.

This cyclical framework gives a convenient reference point for discussion, but it is an abstraction of the overall process. Of course, the stages of this action-perception cycle interact. For example, the type of data abstraction performed by the system may be influenced by the current plan of action, or the actions of the system will affect the types of data available for interpretation. Each stage interacts with all others. Also, the timing of the stages is not necessarily a sequence of: sense, abstract, interpret, plan, act. The actions of each stage can be interleaved with those of others. The structures of interaction and timing relations among these stages are precisely the issues that we wish to study.

From a slightly different view, the problems of interpretation are those of abstracting sensed events through several levels of complexity and expanding intended actions through several levels of detail. Selective attention helps in both the data reduction process and the process of selection actions. It is the manipulation of that selective attention that is a core issue.

Machine vision research is the study of an artificial system. Each such system results from a process in which designs are proposed, implemented, tested, and refined. From our experience with these artificial systems, we come to understand the ways in which the system can be made to work. We are, also, able to induce theoretical frameworks and structures that allow us to predict the possible actions of systems of future designs. In short, design principles emerge from both experience and theory. Because of this, the exploration of alternate solutions to interpretation is essential.

To summarize, we are building complex systems ~~in~~ which errorful data, locally noisy processing, and tentative hypothese are combined to produce globally robust and certain results. To achieve this, we develop algorithms which apply combinations of knowledge and processing in an opportunistic fashion. This is a process of successive design and testing, so we want the basic system to be flexible enough to allow for the rapid exploration of alternative designs.

Two Knowledge-Based Interpretation Projects

In order to develop an understanding of knowledge-based control, we are investigating two areas. The first is knowledge-based planning of camera motion and the second is knowledge-based object tracking; roughly, these correspond in the study of human vision to the notion of eye movement and attention, respectively. These two projects share a common set of goals. In addition to developing methods for reducing uncertainty by combining partial interpretations, we are interested in understanding how to construct an interpretation system so that it can operate in real time. Our current goals do not include the construction of an actual real-time system. Rather, we are concentrating on developing methods of interpretation that are robust in the face of changes in data. Such methods can the be applied to partially interpret a particular frame, gathering as much information as allowed by the computing researches allocated to that task, and then continue that interpretation on the next frame, updating the interpretation as it goes. Thus, the primary goal, is understanding the issues of selection and attention to data and processing within the framework of interpretation. In studying the control of interpretation processes and camera motion in these domains, we will increase our understanding of methods for interpretation of dynamic scenes.

The Active-Eye(knowledge-based camera motion)

We chose to study the use of knowledge in controlling the camera because this problem presents the simplest closed-loop system for scene interpretation. Each image is a probing of the environment; the information extracted from the image represents a sample of the environment. The partial interpretation that results from that sample gives rise to new goals, some of which require additional information. Parts of the newly needed information cannot be derived or inferred from the current image; a new image is required. Determining the best position for acquiring the new image and selecting the procedures for extracting the new information from that image are control problems, which results in a plan and a series of actions changing the position of the camera in the environment.

Changes of camera position are controlled by the interpretation system: thus the relation between the previous interpretation and the new image can be predicted. With these predictions, the system can combine the additional information (a new set of partial results) with the previous results to increase the scope and detail of the interpretation. It is the understanding of this process of the collection of partial results (over time) that is the object of this study.

Three major issues are being investigated. First, there is a trade-off between data-driven processing of each image–in which much computation is spent in trying to derive surface shape and object boundaries from image characteristics–and knowledge-directed processing–in which particular features are selected to confirm expectations, satisfy goals, and enable extension of existing hypotheses. Second, there is the choice of searching among possible interpretations using the current data and searching within the environment for new information to confirm or refute current interpretations. Finally, there are questions on how to integrate hypotheses from many levels of abstraction, especially those that are qualitative in nature with those (as dealing with position information) that are quantitative.

The domain of investigation in the Active-Eye project is a table-top robot work-station in which objects are placed. Knowledge of the environment and the possible objects and object types is given to the system. The environment is assumed to be static.

Knowledge-Based Tracking

In this project objects of known type and shape are moving in the environment and the system is to segment, identify, and predict the path of each object. As with the active eye project, partial results from each frame are combined to produce a more complete and detailed interpretation as time goes on. Since the interpretation at any time must now include the predicted path of the object, the interpretation of the scene is predictive; it can be used to hypothesize where objects will be in future frames. Thus, the partial results of previous interpretations not only are combined with the current interpretations, but must, in fact, be integrated.

The investigation of the methods for integration of partial results is the key issue in this study. Each set of partial interpretation may be in error. Uncertainties are introduced from the data and by the processing to extract information from that data. When combining these partial interpretations conflicts arise, resulting in a need for additional information that serves to focus the processing of subsequent images. Thus, under the control of the interpretation system, processes are abstracting particular information from each image in order to correct, extend, or complete an ongoing interpretation of the dynamic scene.

This study is being carried out in the three domains of increasing complexity: in the first, a robot table-top world containing regular geometric solids is animated (in short, it is a dynamic blocks world); in the second, the system will interpret sequences of images from walking the halls; and in the third, the system will interpret sidewalk scenes.

Summary

Both the camera motion and object tracking systems are being developed to study the issues of combining partial interpretations to reduce uncertainty. In the active-eye project, the partial results are obtained from a system-controlled camera. This permits more control over that manner in which results are combined. In the knowledge-based tracking project, partial results must be integrated; here, issues of knowledge-controlled conflict resolution are being studied. In both these projects we are investigating the issues of how to effectively combine qualitative and quantitative information; of how to match symbolic and geometric descriptions in object models to image data; and how to accumulate partial results in such a way as to reduce uncertainty.

2.2 Planning and Programming

During the past year a continuation of previous work has been carried out in three areas: (1) Path Planning, (2) Graphic Programming of Robot, and (3) Heuristic Problem Solving. Work in the first two of these areas has been completed and work is being initiated in four new areas:

- Assembly Planning,
- Sensor Planning,
- Tolerancing, and
- Development of an Engineering Database/knowledge Management System.

In the following sections we will describe the status of the work in each of these areas. In the new areas, the emphasis will be on the approach being taken and the general nature of the results expected.

2.2.1 Collision-Free Path Planning in a Multiple Robot System

Investigator: K. G. Shin

Problem and Past Efforts

The goal of this research is to develop a collision-free path planner in a multiple robot system. Previous works in this area either consider a single robot system with fixed [1], [2] obstacles, or are too complex to implement in 3D space [3]. The main reason for the complexity is the infinite number of possible paths. One way of reducing this infinite number of paths to a finite number is to divide the workspace. The most common representation used for dividing the workspace is the Octree [4] space representation, in which a workspace is represented by a rooted tree. The root of the tree represents the whole workspace which is a cube. A cube is said to be *homogeneous* if it is empty or occupied by object(s). If a cube is not homogeneous, it is partitioned into eight identical subcubes, each of which is represented as a child of the node representing the original cube. This decomposition continues until all the cubes become homogeneous or the cube's size reaches a given limit. The main advantage of the Octree space is that it limits the search space to a finite space and enables a fast search for real-time implementation.

There are some difficulties, however, in direct implementation of the Octree for a multi-robot system. If a robot moves into a large free cube, the entire cube becomes unavailable to other robots unless the cube is further decomposed in real-time. Thus, it is not economical in utilizing the workspace if there are only a few large cubes. To overcome this difficulty, we use a modified version of the Octree to represent the workspace. The workspace is partitioned into a set of cubes of the fixed size (called *base cubes*) and each base cube could become a root of a tree. A base cube will be decomposed further until it becomes homogeneous or the decomposition reaches a certain limit the cube's size.

Status of the Research

It is assumed that each robot can either *advance* its end-effector to a neighboring base cube of the current position or *wait*. The output of a path planner only specifies the precedence constraints, such as 'robot 1 should reach cube V_i before robot 2 reaches cube V_j,' that have to be met by the trajectory planner. More on this will be discussed later.

Another question to be answered in the above assumption is 'what if there is no path from the origin to the destination by connecting base cubes?'. When a base cube is not available, the path planner searches for a subcube of the base cube that will lead to a collision-free path and the path planner will assume that the base cube has been temporarily moved to a *pseudo-base cube* at the center of the sub-cube. The pseudo-base cube is treated the same as a normal base cube. This pseudo-base cube and the obstacle that causes the movement of the base cube will be stored in the system as long as the obstacle's position remains unchanged.

Under the Octree description of the workspace, the path planning problem becomes as follows:

Problem: Find a minimum number of (pseudo-) base cubes for a given origin-destination pair of each robot and precedence constraints to avoid collision. Note that the cube associated with a "wait" operation will be counted as many times as some quantity proportional to the wait time. Henceforth, the number of cubes will be referred to as *distance*.

A robot moving from a cube to another is said to *trespass* a set of cubes when the sweeping volume of the movement intersects every cube in the set. Collision occurs when a robot attempts to trespass a set of cubes that is not disjoint with either of the

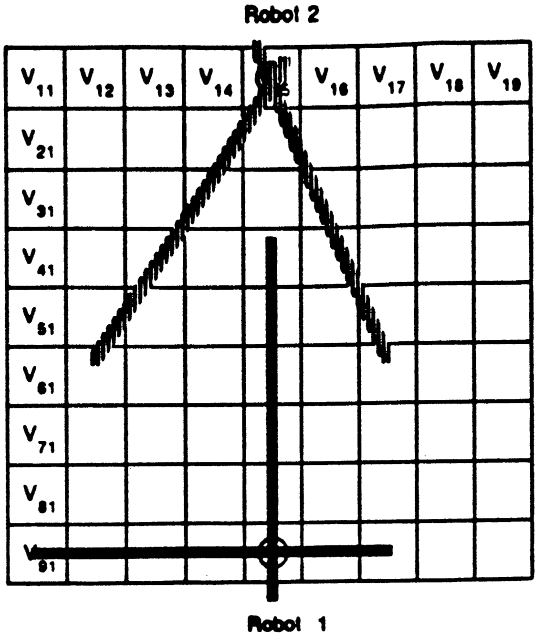


Figure 4: Example of timing constraint.

following:

1. The set of cubes that are occupied by some fixed obstacles.
2. The set of cubes that are trespassed by the other robot.

In case of (1), the robot has to detour or a pseudo-base cube is generated, if necessary. In case of (2), the robot may have to detour or precedence constraints will be added. Figure 4 shows such an example in 2D space. While robot 1 can follow the sequence of cubes $V_{91}V_{82}V_{73}V_{64}V_{55}V_{46}$, robot 2 cannot follow $V_{67}V_{66}V_{65}V_{64}V_{63}V_{62}$ because of the crossover between the two paths. Without precedence constraints, robot 2 has to follow $V_{67}V_{57}V_{36}V_{35}V_{44}V_{53}V_{62}$ which is not optimal. This kind of situation arises due to the lack of timing information in the path planning stage. With precise timing information, we may allow robot 2 to follow $V_{67}V_{66}V_{65}V_{64}V_{63}V_{62}$ as long as robot 1 does not reach V_{73} until robot 2 reaches V_{63}.

However, the exact timing information can only be obtained by a trajectory planner after paths have been determined, and obtaining them is very time-consuming. Instead of obtaining them from the trajectory, our path planner provides precedence or timing constraints to be met by the trajectory planner. A timing relation $< |_p$ is defined as follows:

Definition: A cube V_1 is said to be 'less than' a cube V_2 under a pair of paths $p = (p_1, p_2)$ denoted by

$$V_1 \; <|_p V_2 \; iff$$

the following hold.

1. V_1 is in p_1, the path of the end-effect of robot 1, and V_2 is in p_2, the path of the end-effector of robot 2, or vice versa.
2. The robot should pass V_1 before the other robot reaches V_2 to avoid collision between them.

The *best first search* is used:

$$h(p) = max(h_1(p), h_2(p))$$

where $h_i(p)$ is the sum of $f_i(p)$, the number of cubes trespassed by the end-effector of robot i, and $g_i(p)$, the projected number of cubes that will be trespassed by it to reach the destination. The goal of the search is to find a path from the origin to the destination that minimizes $h(s)$ for each robot.

If collision is detected during the search, a robot advances while the other waits. Choosing a robot that will advance is based on $h_i(s)$. This procedure continues until a robot clears ways for the other. This can be achieved by moving a robot away or expanding another path that was not attractive could become more attractive because of a long wait. The timing relation is established in the first case.

Figure 5 shows the search tree generated by the case shown in the Figure 4.

Future Research

- Develop an improved heuristic estimator of distance that will reduce the number of search steps.
- Develop a trajectory planner that will work with the path and timing timing constraints generated by this path planner.

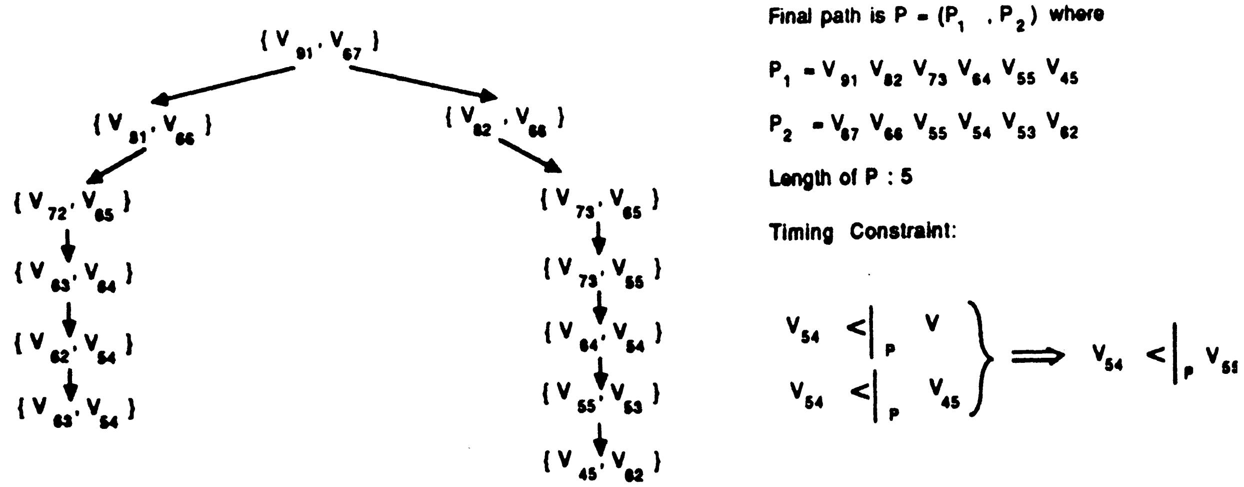


Figure 5: Search tree generated by Fig. 4

References

[1] T. Lozano-Perez. "Spatial Planning: A Configuration Space Approach." *IEEE Trans. Comp*, February 1983.

[2] R. A. Brooks. "Planning Collision-Free Motions for Pick-and-Place Operations." *Int'l Journal of Robotics Research.* Vol. 2, pp. 19-44, Winter 1983.

[3] D. Johnson and E. Gilbert. "Minimum Time Robot Path Planning in the Presence of Obstacles." *Proc. of 24rh Conf. on Decision and Control.* pp. 1748-1754 December 1985.

[4] C.L. Jackins and S. L Tanimoto. "Oct-Trees and Their Use in Representing Three-Dimensional Objects." *Proc. of Computer Graphics and Image Processing 14*. pp. 249-270, 1980.

2.2.2 Graphical Off-line Robot Programming

Investigators: A.W. Naylor, R.A. Vols, S. Lejun

Research Objective

Graphical off-line robot programming has several obvious advantages [1], [2], [3] over on-line robot programming in that it does not require production shutdowns, debugging is easy and inexpensive, and program development will not lead to the possibility of potential equipment damage. All of these advantages make this method quite attractive. However in order for this technique to be acceptable by industry, a graphical off-line robot programming system should satisfy the following conditions:

1. The system must be able to produce all the statements a robot program language has. The categories of statements include robot movement statement, conditional statement, looping statement, I/O statement, case statement, procedure or function call statement and so on.

2. The system should be easy to use. That is, there should exist a user-friendly interface to the user, providing several viewing and object manipulation methods so that user can identify the positions of robot or other objects and control their movements on graphics screen easily, etc.

3. The system should not only be able to simulate the robot's movement on graphics screen, it should also be able to simulate any changes of the environment. The simulation of the program should be in real-time, and pictures displayed should give the user the impression that he seems in the real environment.

The research objective is to develop a graphical off-line robot programming system which satisfies all the conditions.

Status of the Research

During the past year, two major advances have been achieved. These are:

- The entire system has been reorgnised to be more efficient and provide interactive debugging utilizing a simulator.

- The concept of simulation language and simulation program have been introduced.

System Reorganization The original goal of the system redesign was to make the system flexible, expandable and to increase its capabilities. At present time the redesign

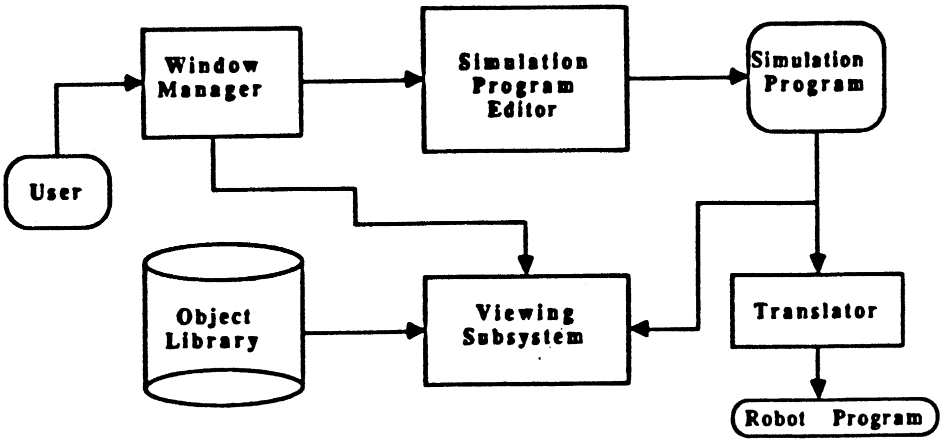


Figure 6:

phase has been completed. The redesign is based upon a careful separation and modularization of both data representation and functions of the programming system and consists of five components. They are: Window Manger, Simulation Program Editor, Translator, Object Library and Viewing Subsystem. Its function diagram and interconnections are shown in Figure 6. Each component is implemented by one or more modules. If we want to modify or enhance the functions of some component, all we need to do is to modify the corresponding module without affecting other parts of the system. For example, if we want to program another robot model using our system, we only need to replace the robot module by another one which can implement all the functions of the new robot model and insert a new object item which describes the geometric representation of the new robot model into the Object Library. This feature makes the system very flexible and easily expandable.

The functions of each component except Simulation Program Editor were described in last year's report and will not be discussed further here. A more complete description of the systems can be found in [4]. The Simulation Program Editor can accept various kinds of program-editing commands sent by the user through selecting desired menu items from corresponding window to construct, edit and simulate robot programs. Possible editing actions include: Insert a statement into a program; Delete a statement from program; Modify any statement in a program; and Simulate a program on graphics screen. In addition, it has a special mechanism used to detect and prevent some other errors which in general can only be found at run time.

The menu system is a part of the Window Manager and has been redesigned to make it easier and clearer to use. Based on the functions, menus are divided into two groups, a program editing menu group and a viewing specification menu group. Each group consists of a set of submenus. Each submenu forms a menuframe which consists of a list of menu items. Program editing menus are used by the user to develop the simulation program, whereas the viewing specification menu are used to select proper viewpoints to view the current environment on graphics screen. The two groups of menus are independent. However within each group, its submenus form a hierarchical structure. At any time two menu frames, one from each group, are active. But only one frame at a time appears on the window. The user can decide which one is to be displayed by toggling a button.

Simulation Program

Logically, the programming process in our graphical programming system can be described briefly as follows:

1. First, the user creates a graphical simulation program.

2. The program is then simulated on the graphics screen .

3. If not satisfied, the user can modify the program and repeat step 2.

4. The simulation program is translated into a workable robot program.

The functions of a graphical simulation program are different from those of a robot program in that a robot program needs only describe how a robot moves; but in addition, the simulation program must be able to describe the environment a robot is working in and any changes of the environment. Thus, both the robot's movements and the current environment should be shown on the graphics screen as the simulation program is running.

The actual programming process is a mixture of menu selection and graphic operations. For example, consider the process of a robot repeatedly picking up parts of the same type on an assembly line. Each part first moves to a certain location along the assembly line. Then the robot picks it up and finally puts it down to another location. In order to simulate this process on graphics screen, the corresponding textural part of the simulation program might look as follows:

```
CREATE OBJECT - ASSEMBLY LINE -      a part of the robot's environment

REPEAT
        CREATE OBJECT - PART
        MOVE OBJECT - PART
        MOVE ROBOT and PICK UP OBJECT - PART
        MOVE ROBOT and PUT DOWN OBJECT - PART
        DELETE OBJECT - PART
UNTIL < condition >
```

Each of the "statements" is created from a menu selection. The "MOVE" statements also invoke a graphics mode whereby the programmer, under joystick control, directs the movement of the graphical representation of the object under consideration.

In general the graphical simulation language should contain at least three different types of statements:

TYPE1, TYPE2 and TYPE3 statements.

All of the TYPE1 statements are robot-related statements and often involve joystick operations on the graphical representation of the robot. Each TYPE1 statement can be translated into some robot language statement. This kind of statement can be used both for simulation purpose and for emitting the actual robot program. The TYPE2 statements have no relation to robot motion. Their functions is to describe the changes of environment and are used solely for simulation purpose. From the short simulation project just given, we can see the usefulness of TYPE2 statements. A complete list of TYPE1 and TYPE2 statements used in our system are given in Table 1 and 2 respectively.

(1) Robot Related Statement:	Move Robot Set Speed Set Configuration Open Gripper Close Gripper
(2) Loop Statement:	Repeat ... Until <cond.> While <cont.> Do ... End
(3) Condition Statement:	If <con.> Then ... End
(4) I/O Statement	Set Actuator
(5) Naming Location Statement:	Name <location> As <identified>

Table 1. A listing of TYPE1 Statements

(1) 3-D Object Related Statement:	Create Object Position Object Select Object Delete Object Name Object Location Modify Object Parent
(2) 2-D Object(I/O) Related Statement:	Create I/O Position I/O Select I/O Delete I/O

Table 2. A listing of TYPE2 Statements

TYPE3 statements are also used for simulation purpose. They provide viewing-description and time-specification statements etc. At present, rather than provide explicit TYPE3 statements, the viewing information is provided by the user as needed via joystick control.

Another important consideration in designing a graphical simulation programming language is the extent of its applicability. One can take two approaches. First, it could be designed to support a particular robot programming language. In general, such a system can only be used with a specific robot model. The second approach is to design a system which can support several different robot models. It uses one common simulation programming language and provides one language translator for each robot programming language (analogous to NC machine tool postprocessors). In this case the simulation language should be powerful enough for general use. Our system is generally the second type, but at present the only postprocessor is for the VAL II robot language [5], [3].

Future Research

This project is now complete.

References

[1] F.A. Kelly and M.P. Deisenroth. "Mechanics and The Graphical Off-Line Programming of Robots," *Proc. of the 1984 Int'l Computers in Engineering Conference and Exhibit,* pp. 101-106.

[2] M.S. Pickett, et al. "Robotteach: An Off-Line Programming System Based on GMsolid," *General Motors Research Publication,* 1983, GMR-4465.

[3] S.D. Chawla and W.A. Gruver. "Off-Line Robot Programming with an Integrated Graphics Subsystem," *Proc. of 1984 Int'l Computers in Engineering Conference and Exhibit,* pp. 111-114.

[4] A. Naylor, R. Volz, L. Shao, R. Jungclas, P. Bixel and K. Lloyd. "PROGRESS - A Graphical Robot Programming System," submitted to *IEEE Int'l Robotics Conference,* 1986.

[5] W.A. Gruver. "Industrial Robot Programming Lanugages," *IEEE Transactions on Systems, Man and Cybernects,* July/August 1984, pp. 565-570.

2.2.3 Automated Robot Task Planning

Investigator: K.B. Irani

Introduction

The research in the past year for automated robot task planning has been concentrated on the development of a general and effective planning methodology.

By virtue of its nature, robot task planning can be conceived as special problem solving. When given a goal, the robot needs to automatically synthesize a sequence of actions which, when executed, can bring its "world" from an initial state to a desired goal state. Previously, we developed a methodology for general problem solving. With this methodology, problems can be modeled systematically using mathematical formalism. The methodology also provides a domain independent mechanism for automatically generating admissible search heuristics which guarantees the optimal problem solution, if one exists. The methodology has been successful when applied to a set of problems. However, it appears to be inefficient for solving robot task planning problems which are characterized by their large search spaces, inherent mutual constraints among subgoals, and incomplete goal specifications.

In view of this, the research in the past year was directed towards the extension and modification of the problem solving methodology which can handle robot task planning problems effectively, while preserving generality and other nice properties. We have mainly achieved (1) a systematic approach for detecting the mutual subgoal constraints and a procedure for ordering subgoals in compliance with those constraints, (2) a procedure for automatically augmenting problem goal specifications to make them more complete, (3) the definition of the new heuristic estimation function into which the subgoal ordering constraints are incorporated. These approaches and procedures have all been tested by a few robot planning examples as well as justified mathematically. In the following sections, our research is further detailed.

Ordering Subgoals

A problem goal is usually composed of a conjunction of subgoals, each of which specifies a desired status for a problem object with respect to certain aspect. It is often the case in solving a problem that one subgoal has to fulfilled before the other. This types of mutual subgoal ordering constraints usually exists implicitly in a problem formulation. However, they play an important role in determining the final problem solution sequence. If these constraints are detected, then the subgoals can be ordered and the search can be guided by the subgoal ordering sequence in achieving the goal.

This makes for an efficient search.

Subgoal ordering has been a problem solving technique employed to cut down search in many early systems. However, the method for subgoal ordering has not been developed satisfactorily. The subgoal ordering strategies in the early systems either make too strong initial commitment, causing considerable backtracking, or make too late commitment, resulting in much redundancy as well as the necessity for detecting and handling conflicts.

In our approach to subgoal ordering, we first define relations which can identify the mutual constraints between pairs of problem subgoals with respect to the relative order of their achievement in any problem solution. If we represent two subgoals as g_A and g_B, then the constraint which can be identified between the two subgoals is either "g_A must precede g_B in any problem solution" or "g_A must not precede g_B in any problem solution" or "g_A and $g - B$ can(cannot) be simultaneously achieved in any problem solution". The relations identifying these constraints can be constructed solely by the knowledge of problem rule specifications. A procedure for ordering a set of subgoals is developed which arranges the subgoals into an order complying with the detected constraints. The result of applying this procedure to a set of subgoals of a goal is an ordered sequence. Each element in the sequence is a conjunction of a subset of subgoals. If the problem subgoals are achieved in the order of their positions in the sequence, the will not be violated once they have been realised. In other words, if this ordering constraint is observed, then no reordering of problem subgoals will be necessary in the problem solving process and therefore backtracking can be reduced.

Goal augmentation

In a problem formulation, goals are often not specified completely in the sense that many things are left as "don't care". In such cases, many states are candidate goal states. However, as far as the optimality is concerned, only a few of them can really qualify to be the candidate goal states. Since in the heuristic search, the cost for the shortest path between any generated state and the closest candidate goal state is to be evaluated in the heuristic estimation, it is obvious that smaller the set of candidate goal states is, the more accurate is the heuristic estimation. Therefore, it is desirable for the goal specification to be as complete as possible.

In the past year, we developed a procedure which can augment the specification of a goal to make it complete. Again, the augmentation depends only on the rule formulations. It is proved that this augmentation does not eliminate any existing optimal problem solution while improving upon the quality of heuristic estimation. This technique is used to augment each component in the ordered subgoal sequence generated by subgoal ordering procedure so that tighter heuristic estimation can be achieved.

Heuristic Estimation with Subgoal Ordering Constraints

Heuristic estimation is an important means to control the search direction in the problem space. Its tightness or accuracy in evaluating the cost of the optimal path from any non-goal state to the goal set has very strong effects on the efficiency of the search. The heuristic estimation function developed in our previous problem solving methodology does not make any commitment to subgoal ordering. As an extension to our previous work, we developed a new heuristic estimation function, into which the knowledge of subgoal ordering constraints is incorporated. This function is still domain independent and admissible. However, it provides tighter heuristic estimation than the previous heuristic, which means that the value of the new heuristic for any non-goal state is more accurate and therefore is more powerful in reducing search for an optimal problem solution. These properties have been proved theoretically and tested by simple examples in the robot task planning domain.

2.2.4 Automatic Assembly Planning

Investigators: J. Wolter, A.C. Woo, R.A. Volz

Problem Definition

In recent years, several authors have proposed robot programming languages in which actions are described in terms of how parts are to be moved, instead of how the robot is to be moved. Such languages, which are known as "task-level" languages, would be much simpler to program in than any currently available languages. Large amounts of research in path planning, fine-motion planning, grasp planning, and automatic error recovery are now being done toward this end. As task-level programming languages begin to become a reality, a new question may be considered: If generating programs in task-level languages is so simple, can we construct a computer system to generate them automatically? Ideally such a system would be fed a geometric model of an assembly from a computer-aided design system, and would produce a task-level program to build that assembly from its component parts.

Such an assembly plan would consist of an ordered list of assembly steps, each of which specifies a set of parts which are to be moved into position relative to a set of fixed parts, and the trajectory along which those parts are to be moved. To complement this, specifications for the fixtures which are to hold the fixed parts must be generated. Since there will normally be many different assembly plans possible, there must be some basis to choose the best plan. Many criteria for evaluating plans are possible. Four of the most obvious ones are:

1. Number of operations.

2. Uniformity of operations.

3. Number of fixtures.

4. Complexity of fixtures.

The relative importance of these (and other) criteria are largely dependent on the application. For example, a different plan may be preferred depending on whether a large or small number of assemblies will be built, or whether they are to be built on an assembly line or in a single manufacturing cell.

Approach

Because of the large number of plans which may be possible and the complexity of the criteria for evaluating plans, an artificial intelligence approach will be used. The effectiveness of such an approach depends largely on being able to view the problem at a fairly high level, so that it is possible to make decision without working out a plan in full detail.

To achieve this, we plan to learn to recognize common structures in assemblies for which assembly strategies are known, and then combine the assembly plans for those structures into a plan for the complete assembly. In mechanical assemblies, the most common and useful such structure is the generalized screw.

A generalized screw consists simply of a shaft with a sequence of parts on it. Bolts, axles, and cotter pins all fit this model. Though there are many possible ways to assemble such a structure, there are only two plans in which all insertions are done from the same side. Either the parts are placed one-by-one on the screw, or the parts are stacked in a fixture, and the screw is inserted last. The choice of which way to build the screws can be made at an early stage. For example, we may prefer placing the parts on the screw because this generally requires a simpler fixture, or we may prefer to construct all screws with parallel axes from the same direction.

Having made an initial decision on how each screw is to be assembled, we must then attempt to combine the plans for each screw in a plan for the entire assembly. There are two ways in which a pair of partial plans may be combined. They may either be done in parallel by merging the plans, or they may be done serially by building one and using it as a subassembly in the other. When there is more than one choice, we must be guided by our decision criteria. Generally merging is preferred, since each subassembly requires an additional fixture.

In many cases it may prove impossible to combine a set of crew plans. In such cases it is necessary to backtrack and try some alternative screw plans. Even then, it may be that the assembly cannot be constructed by moving parts only parallel to the axis of screws, however, even in such cases this approach will generate a partial plan for those portions of the assembly that can be built in such a way, thus reducing the size of the problem which would have to be handled by a more general algorithm.

Plans generated in this way tell us only which parts must be inserted before a given part can be inserted. That is, it would not tell in what order to insert the bolts. Thus there must be a final linearization step to produce a specific assembly program.

Status

This research is still in its early stages. Initial investigation of the characterists of the search space and the implementation of the algorithms to combine partial plans has begun.

Calculations have been performed to estimate the number of possible ways in which simple assemblies can be built. If we consider an assembly containing seven screws each of which can be built only in one of the two preferred ways, then there are in excess of 5×10^{106} ways in which the plans may be combined. When we consider that there are actually 2^n ways in which a screw with n parts on it can be built, and that there are many ways in which each plan can be linearised, it is clear that any kind of exhaustive search of the solution space is out of the question even with very simple assemblies.

Study is now in progress on the representation of partial assembly plans and on how to manipulate them. We must be able to systematically list all the ways in which a screw plan can be merged into an existing plan. This will form the basis of the search strategy.

Future

During the next year an experimental implementation of this assembly planning system will be constructed. Simple interfaces to a geometric modeller and a simulated robot cell will likely be included. In addition, some investigation of more general planners will be done, especially with regard to how they can be used to assist a heuristic system when it runs into problems.

References

Jan D. Wolter, "Automatic Assembly Planning," dissertation proposal, University of Michigan, October 1985.

2.2.5 Sensor Planning

Investigator: R.A. Volz, A.C. Woo, R.Desai, J. Xiao

Problem Statement

The problem of robot motion planning for automatic mechanical assembly can be roughly divided into three sub problems, global motion, grasp planning, and local motion. Global motion is characterised by relatively large movements of robot joints to move the robot from a starting point to a destination point in the workspace [1,2,3,4]. Grasp planning refers to how an object should be grasped for a stable prehension [5,6,7,8,9]. Local (fine) motion, on other, hand involves relatively small movements of robot joints in order to (in case of a mechanical assembly) effect a mating of a held part with another part or parts in the workspace [10,11,12]. It is local motion that is of interest to us here.

Local motion planning is a very complex problem. Some of the factors contributing to this complexity are:

1. Uncertainty in robot motion.
2. The CAD model of the world and the actual part geometry are never in complete agreement regarding dimensions, due to manufacturing limitations, resulting in geometric uncertainty.
3. Errors associated with the sensed data.

We divide the local motion and control problem hierarchically into two levels, generation of a nominal plan (which is generated off-line) and an on-line *verification* and *replanning* system. It is the latter level that we are concerned with here. A conceptual diagram of the system is shown in Figure 7.

Approach

We start by assuming that the errors in the sensors are bounded, and, that curved surfaces can be approximated by polyhedra. The geometric state of an assembly system at any fixed instant of time can be described by vector X consisting of the robot

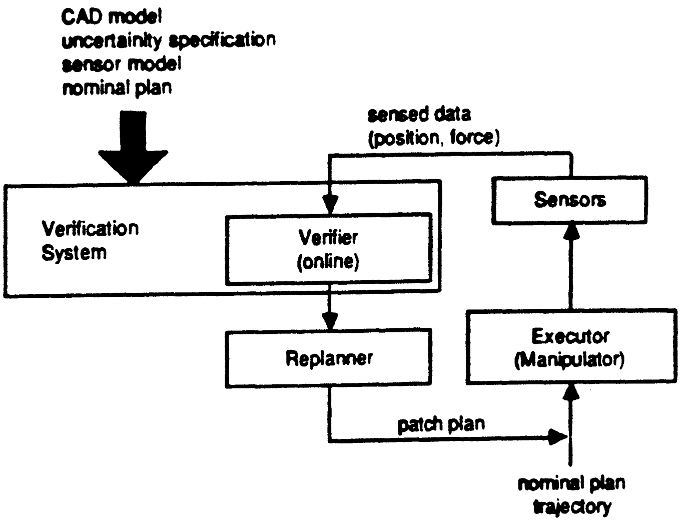


Figure 7: Overall Structure of the Suggested system

joint positions and geometric transformations for every object in the system from a fixed reference point. This is termed *configuration* of system [3,13]. The sequence of configurations that takes the assembly from initial configuration to goal configuration called an *assembly plan.*

A local motion system is introduced which takes as an input a sequence of configurations. In our model, each planned configuration is converted into the commanded joint transformation [T], and a set of guards on the motion [G]. Guards are a set of conditions for sensors. When any member of set of guards is satisfied the commanded motion terminates. Then, the motion command can be written as,

$$\text{MOVE } [T]\ [G]$$

As the nominal local motion plan is executed, some planned configurations will not be achieved due to the positional, geometric and gripping uncertainty. Instead a new configuration is reached. The system we propose would identify the contact formation (defined below) actually reached and in event of a failure, replan the motion to resume the overall assembly task. The overall structure of the proposed system is shown in Figure 7.

We assume that position and force sensors are available. It is apparent that force sensing is useful only for configurations involving contact among parts in the workspace. Therefore, we restrict our verification to that of configurations involving contacts.

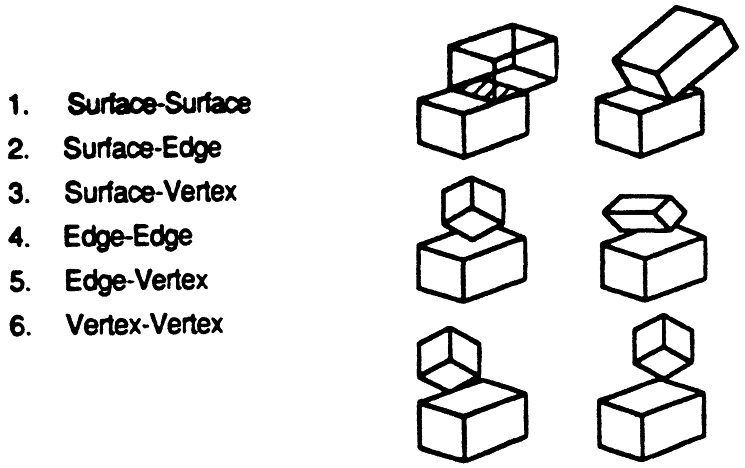


Figure 8: Elemental Contacts

Contact Formations

We introduce a concept of Contact Formations, by use of which we can reduce the complexity of the search problem. Also, contact formations have implicit in them the information regarding the relative degrees of freedom of the contacting objects and therefore the local compliant behaviour (i.e., allowable directions of motions) of the contacting objects. This is very important for successful replanning.

The geometrical model of a part consists of three topological elements, surfaces (faces in case of polyhedral objects), edges and vertices. We define an elemental contact as being a contact between any two topological elements (Figure 8). An elemental contact between topological element e_i^l of object i and topological element e_j^k of object j is represented as a pair $\{e_i^l, e_j^k\}$.

Next, we define a *contact formation* as a set of all configurations, with identical elemental contacts between the same pairs of topological elements. We can also view a contact formation between objects i and j as a set of elemental contacts,

$CF = \{.., \{e_i^l, e_j^k\}, ..\}$. By slight abuse of notation, we denote contact formation r by CF_r regardless of whether we view it as being composed of a set of configurations or a set of elemental contacts, and write $\{e_i^k, e_j^l\} \in CF_r$ or configuration $X \in CF_r$, as is convenient. For configurations having no contacts we write $CF = \phi$.

Contact formations partition the configuration space into finite number of partitions. This reduces the search problem in configuration space (which is a continuous space), to a much less complex search problem in Contact Formation Space (*CF-Space*) since if there exists a path from the initial to the final configuration then a path can be found that is through a sequence of neighbouring contact formations. Moreover, the path is entirely made up of elemental motions. The details of the proofs can be found in [13].

The local motion planning problem can be now be restated in context of contact formations. We can think of part mating as moving from an initial contact formation to goal contact formation. As we stated above, the verification system identifies the contact formation that the motion terminates in. The replanning system corrects the motion if necessary based on the contact formation and sensed data.

Verification

Suppose the actual configuration attained when motion terminates is X_a, and, F_s, P_s and M_s are the sensed force, position and moment in this configuration, and F_a, P_a and M_a are the actual force, position and moments. The held object must satisfy the following equilibrium conditions,

$$\overline{F}_a = -\int_A P_c(r,\theta,\phi) d\overline{A} \tag{1}$$

$$\overline{M}_a = -\int_A P_c(r,\theta,\phi)\overline{r} \times d\overline{A} \tag{2}$$

where P_c is the contact pressure and A is the contact area. Let S be a set of all the contact formations $\{CF_1, CF_2, ..., CF_a, ...CF_s\}$ that the MOVE command issued at X_s can terminate in. It can be shown, [13], that testing the equilibrium conditions for any given contact formation can be reduced to testing their satisfaction for a particular single configuration in that contact formation. Assuming no error in sensing or geometry, we could test for each possible contact formation to see if it satisfies the equilibrium conditions for that particular formation. Suppose, $\{CF_{n1}, CF_{n2}, ..., CF_{nr}\}$ configurations satisfy the equilibrium conditions. Then the position sensor would be sufficient to identify the contact formation CF_a, again under the assumption that the sensors and geometry are perfect.

Now, if we consider the case where the sensors are not perfect and the geometrical uncertainty is considered, the equilibrium equations are still valid, assuming that r is the actual r_a. But, the position vector may no longer distinguish among all the contact formations that satisfy the equilibrium conditions. This is because actual positions and dimensions (geometry) are not known. Because of the uncertainty there may be two or more contact formations that satisfy both the equilibrium and position constraints. However this cannot be true for sufficiently large distances between a pair of contact formations that satisfy the equilibrium conditions simultaneously.

Replanning Concept

Assuming the verification system identifies the current contact formation, the next

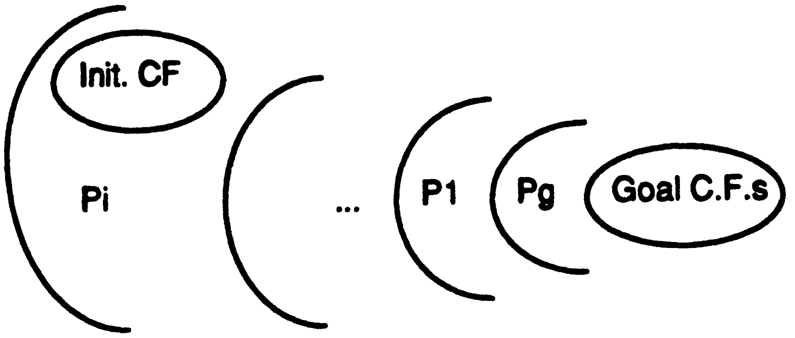


Figure 9: Top-level of pre-image sets

step is to replan the motion of the manipulator, to guarantee the success of reaching the goal. It is introduced in ***three hierarchical levels.***

The top *level* uses the idea of pre-image as described in Lozano-Perez[10]. But the computation of pre-images in configuration space is so complex and the pre-image space is so large that it still remains as an impractical approach. Our approach interprets the pre-image space as a subset of the space contact formations, which is discrete and finite. Hopcroft [14] has shown that if two objects in contact can be moved to another configuration in which they are in contact, then there is a way to move from the first configuration to the second configuration such that the objects remain in contact throughout the motion. By the definition of contact formations, that means for a given initial CF and a goal CF, we can always find a path of CFs between them.

The path in the top level is defined as a sequence of pre-image sets. By using the back-chaining technique, the path can be found conceptually in the following way: starting from the set of the goal C.F.s, expand by *certain motion* to its direct neighbor CFs – those neighbor CFs are then called the *pre-image set* (to be determined) P_g of the goal CFs; continue the expansion from P_g to its pre-image set P_1 and then to $P_2 \ldots$ and so on; stop the expansion when the initial CF is covered by one pre-image set P_j(Figure 9). The top level path (or plan) is just the sequence $P_j, P_{j-1}, \ldots, P_1, P_g$.

The *second level* refines the top level by selecting a path which is a sequence of CFs from the sequence of pre-image set (i.e. the top level path). The key issue here is which neighboring C.F. to move to, when there are more than one valid neighbor. To decide this, various environment factors have to be considered and related graph theory might be applied.

The *third level* deals with the direction of the motion. The direction of the motion may consist of the direction to move from one CF to its neighbor, and the directions to move from one configuration to another within a CF. We assume that only replanning in a small area such that there are no obstacles is considered. Then we may move from one CF to its neighbor or from one configuration to another along a straight line. This simplifies the problem.

Status of the Research

Verifiability and Geometric Restriction on Design

We thus approach the verification problem by seeking design constraints on the objects such that the verification can always be guaranteed. At present our results are restricted to contact formations involving a single contact.

Contact formations are distinguished by considering the change in sensed moment or sensed position for each contact formation. When there is error in the sensed data then it is possible that the change in moment or position is not large enough to be distinguished from the sensed data. This suggests that if we ensure that there is always either a large moment or position change between two contact formations then we can ensure verification. From mechanics and considering the geometry, we develop design restrictions. The analytical expressions for the design restriction can be found in [13].

These conditions translate as simple geometric restrictions on design as in figures 10 and 11.

The geometric restriction manifests [1], [2] itself either as a minimum distance limitation between parallel surfaces or as a constraint on angle between the faces involved in the corresponding contact formations.

We call the angular condition arising from uncertainty as uncertainity cone and positional constraint as uncertainty sphere. It may be noted that for convex objects the only geometric restriction is of the kind where angular restriction is placed on the face normals of the object. The geometric restrictions on design for the case with friction are then similar to the case with no friction except, instead of the uncertainty cone we have a Friction-uncertainty Cone.

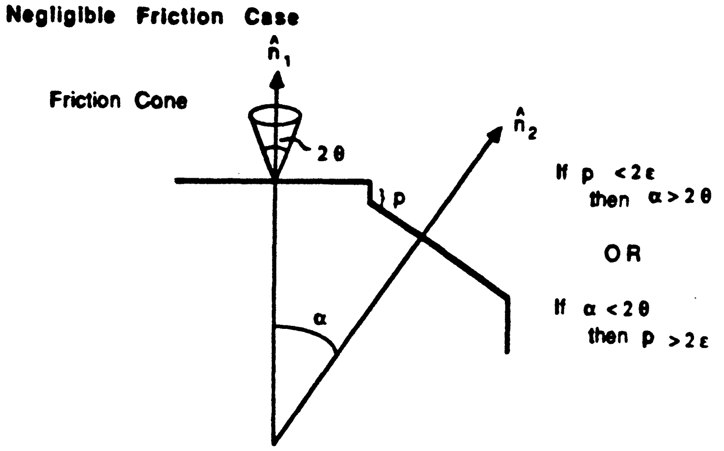


Figure 10: Geometric Restrictions: Negligible Friction

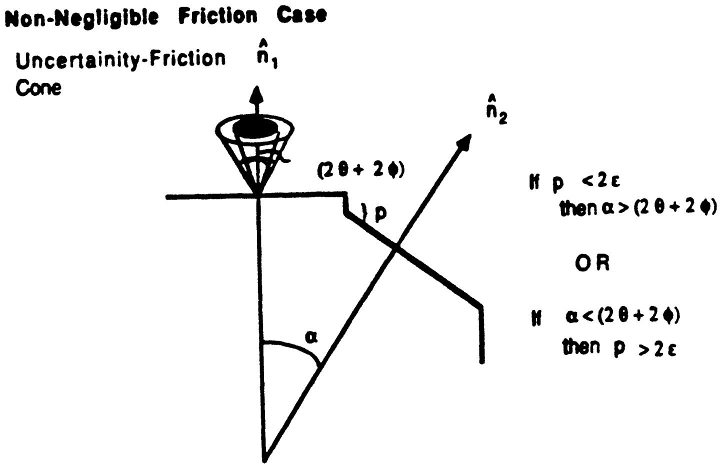


Figure 11: Geometric Restrictions: Non-negligible Friction

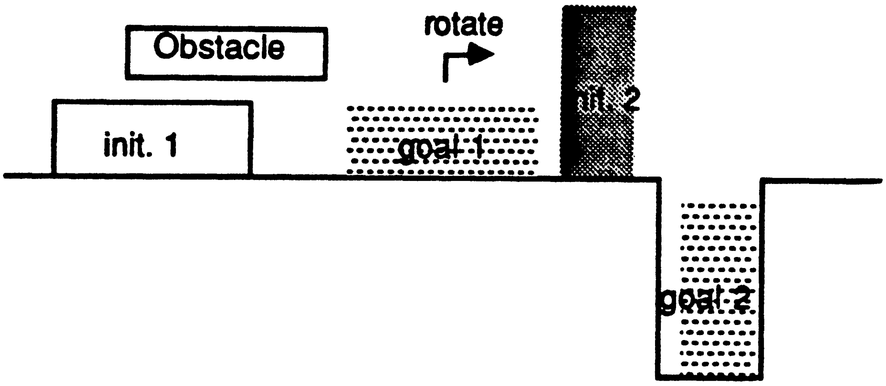


Figure 12: Subgoal Method

We have also developed a hypothesis and testing scheme for determining the set of contact formations which could satisfy the equilibrium equation for given error bounds.

Replanning

Work in the replanning area is only beginning. The first problem here is what kind(s) of the motion(s) is (are) proper for defining the pre-image set of CFs, and for moving within a contact formation . We have defined a *sliding/rotation* model, in which the pre-image set of a CF is its direct neighbors that can be reached by *sliding*(translating) along the surface of contact or rotating about a point. The manipulator should first *rotate* the initial CF to a neighbor on path, then follow a sequence translations. Figure 12 shows a task in a more complex environment. In this case subgoals may have to be selected. For each subgoal, the sliding/rotation model is applied. We are currently investigating design constraints which will allow use to select good directions of motion, in spite of the uncertainties present.

Experimental Testbed

In order to test both the verification of contact formations in the presence of uncertainities and the replanning strategies a testbed is being developed. The first generation testbed is intended for testing operation of our algorithms in 2-D setting. The testbed is sketched in Figure 13. We artificially introduce the sensing and motion errors by adding random numbers to the data received from the sensors and the commanded positions. Currently we are successful in verifying the contact formations that motion

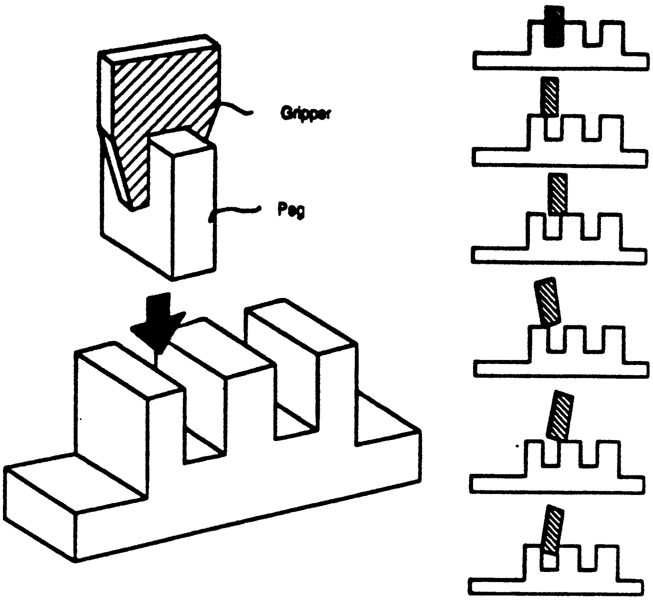


Figure 13: Experimental Testbed

terminates in.

So far we have developed the concept of Contact Formations and some important results concerning it's role in local motion operations. A general structure for a local motion execution system is also introduced. Design rules have been developed which allow disjoint contact formations to be distinguished in spite of uncertainities. An algorithm for distinguishing is developed. A 2-D testbed has been setup and some initial verification results and algorithm have been verified.

There is a class of contact formations (pairs on contact formations whose intersection is not null) that cannot satisfy the design restrictions. Then the verification algorithm discussed above cannot distinguish among them. The verification work is now being extended to cases where it is not possible to meet the design restrictions and therefore the simple verification strategy is not sufficient.

Future Directions

The hierarchical approach towards replanning greatly reduces the dimensionality of the problem space, therefore the complexity of the problem. The top and the middle level replanning strategies apply to both global and local motions. Since we assume a good nominal plan such that the failure of reaching a goal is due to uncertainties only, it is the fine motion we are mostly interested. Thus, we will focus on the third level replanning as future research. We may further investigate related AI techniques, sensor information and other knowledge. The approach will be math model in combination with rule based system technique.

References

[1] E.G. Gilbert and D.W. Johnson. "The application of Distance Function to Optimization of Robot Motion in Presence of Obstacles," *Proc. Conf. Decision and Control,* Las Vegas, Nevada, December 1984.

[2] C.S. Lin and J.Y.S. Luh. "Optimal Path Planning for Mechanical Manipulators," *ASME Jour. Dynamic Systems, Measurement and Control,* Vol. 102, pp. 142-151, June 1981.

[3] Tomas Lozano-Perez and M.A. Wesley. "An Algorithm for Planning collision-free paths among polyhedral obstacles," *Communications of ACM,* Vol 22, No. 10, pp. 560-570, October 1979.

[4] S.M. Udupa. "Collision Detection and Avoidance on Computer Controlled Manipulators," *5th Joint Conf. Artificial Intelligence,* MIT, 1977.

[5] J. W. Jameson. *Analytic Techniques of Automated Grasp,* Department of Mechanical Engineering, Stanford University, June 1985.

[6] M. R. Cutkowsky. *Robot Grasping and Fine Manipulation,* Kluwer Academic Publisher, Boston, MA, 1985.

[7] H. Hanafusa and H. Asada. "Stable prehension by a robot hand with elastic fingers ," *Robot Motion: Planning and Control,* 1982.

[8] J.D. Wolter, R.A. Volz and T.C. Woo. "Automatic generation of gripping positions," *IEEE Trans. Systems, Man and Cybernetics,* Vol. SMC-15, No. 2, pp. 204-213, April, 1985.

[9] M.T. Mason. "Manipulator grasping and pushing operations," *Robot Hands and the Mechanics of Manipulation,* eds. M.T. Mason and J.K. Salisbury, MIT Press, Cambridge, MA, pp. 171-298, 1985.

[10] Tomas Lozano-Perez, M. T. Mason and R. H. Taylor. "Automatic Synthesis of Fine Motion Strategies for Robots," *The Inter. Jour. of Robotics Research,* Vol. 3, No. 1, pp. 3-24, 1984.

[11] M.T. Mason. "Automatic Planning of Fine Motion: Correctness and Completeness," Computer Science Department and Robotics Institute, CMU, October 1983.

[12] M.A. Erdmann. "Using backprojections for fine motion planning with uncertainty," *The Int. Jour. Robotics Research,* Vol. 5, No. 1, pp. 19-45, 1986.

[13] R.S. Desai and R.A. Volz. "Contact Formations - Toward A New Approach To Local Motion In Mechanical Assembly," Robotics Research Laboratory, Department of Electrical Engineering and Computer Science, University Of Michigan, IBM/86, September 1986.

[14] J. E. Hopcroft and Gordon Wilfong. "On the Motion of Objects in Contact," *Technical Report (TR 84-602), Department of Computer Science, Cornell University,* May 1984.

2.2.6 Optimal Tolerance Assignment

Investigator: A.C. Woo

Problem Statement

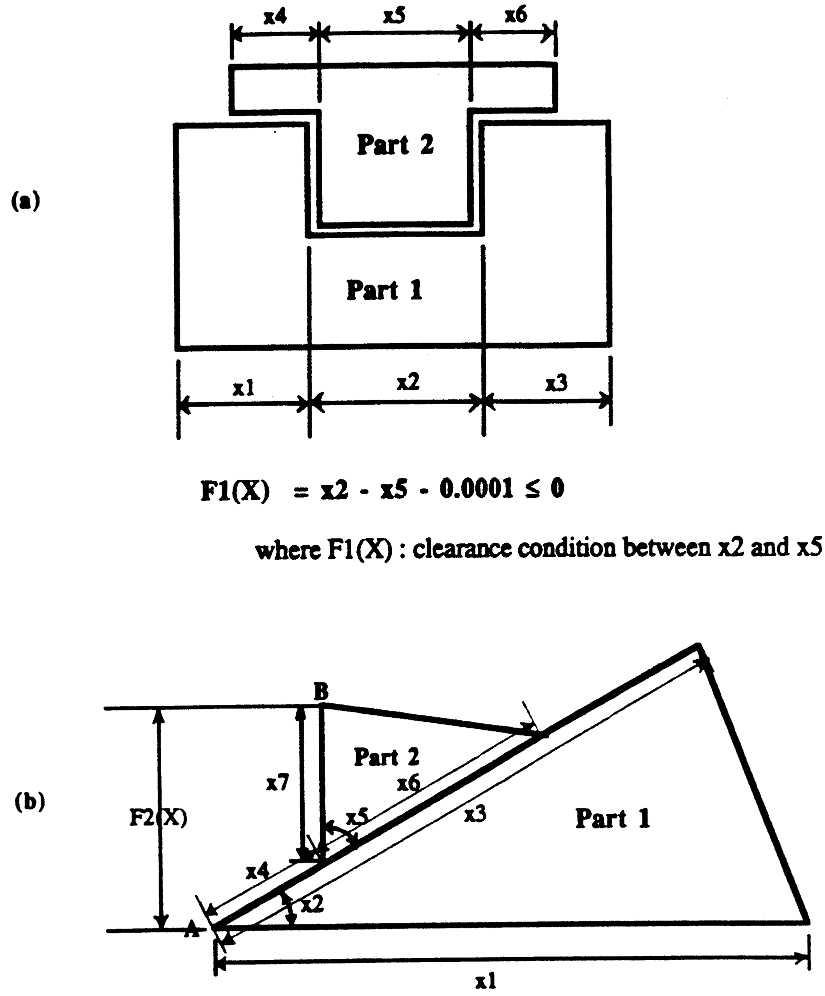
Dimensional tolerance, from the viewpoints of precision in design and performance in function, should be as near to sero as possible. But, because of constraints such as manufacturing cost, larger than ideal tolerance is often assigned. Conversely, tolerance assignment may be too stringent. A looser tolerance leading to a rejection rate of R% may return a cost saving of C%. It is the trade-off of cost and performance of manufactured goods that motivates us to study the problem of optimal tolerance assignment. The result of this study will yield a rational justification for tolerances in mechanical designs.

To be sure, tolerance analysis is practiced today in the industry. It presumes that tolerances have been assigned. Possible conflicts there can be detected through analysis. However, no automated facility exists for arbitrating a conflict. Since our study intends to synthesize tolerances, it subsumes analysis. In other words, our result will be not only free of conflict but also optimal in cost and functionality.

We define functionality by an inequality

$$F_i(x_1, x_2 \ldots x_n) \leq 0 \qquad \text{for } i = 1, 2, \ldots m$$

where $x_1, \ldots x_n$ denote dimensions and m is the number of inequalities. Figure 14 shows two such inequalities, one is linear and the other is non-linear. Given a machining cost



F1(X) = x2 - x5 - 0.0001 ≤ 0

where F1(X) : clearance condition between x2 and x5

F2(X) = x4 sin(x2) + x7 sin(x2+x5) - 5.2000 ≤ 0

where F2(X) : vertical height condition from A to B

Figure 14: Example of Linear and Nonlinear Fundamental Functions

function $C(t_1, t_2 \ldots t_n)$ in terms of tolerances t_i, the tolerance assignment problem can be formulated as follows.

$$\text{Min} \quad C(t_1, t_2, \ldots t_n) \tag{3}$$

subject to $Pr\{F_1(x_1 \ldots x_n) \leq 0, \ldots F_m(x_1 \ldots x_n) \leq 0\} \geq 1 - R$ where $t_i \geq 0$, for $i = 1, \ldots, n$, and R is the rejection rate. By treating a dimension x_i as a random variable with a normal distribution [1], we transform tolerances t_i to standard deviations σ_i. Hence we have

$$\text{Min} \quad C(\sigma_1 \sigma_2 \ldots \sigma_n) \tag{4}$$

subject to $Pr\{F_1(x_1 \ldots x_n) \leq 0, \ldots F_m(x_1 \ldots x_n) \leq 0\} \geq 1 - R$ where $\sigma_i \geq 0$ for $i = 1, \ldots n$.

Approach

Formulation (4) has the following interpretation. Choose a set of standard deviations σ_i for the random vector ($x = (x_1 \ldots x_n)$ such that the cost C is minimized. Intuitively, such a solution corresponds to an ellipsoid of minimum radius inscribed by a set of hypersurfaces. In other words, we wish to find a point $x^* = (x_1{}^* \ldots x_n)^*$ on some hypersurface whose distance to the point $\bar{x} = (\bar{x}_1 \ldots \bar{x}_n)$ is σ 2. But there are two major difficulties. First, the random variables are not independent. Second, the hypersurfaces are not linear.

Hasofer and Lind [2] suggests that the dependency among random variables can be estimated by their average $\bar{x}$ and covariance V. The transformation needed to make the correlated variables independent has three steps: first, translation $x' = x - \bar{x}$; second, orthogonal transform $Y' = P^t x'$, where P is the orthogonal matrix to diagonalise V through $P^t V P = V_y$, third, standardisation $Y = D^{-1} Y'$, where $DD^t = V_y$. The the transformed variable

$$Y = D^{-1} P^t (x - \bar{x}) \tag{5}$$

follows the independent standardised normal distribution.

Using transformation (5), we scale and translate an ellipsoid centered at $\bar{x}$ to a hypersphere centered at the origin. The distance squared from the origin to any point is thus

$$\beta^2 = Y^t Y = (x - \bar{x})^t V^{-1} (x - \bar{x}) \tag{6}$$

Because of the positive definiteness of the covariance matrix V, we can minimise β^2 (rather than β) and formulate our tolerance optimisation problem as:

$$\begin{array}{ll} \text{Min} & \beta^2 = (x - \bar{x})^t V^{-1} (x - \bar{x}) \\ \text{subject to} & F_i(x) \geq 0, \quad for\ i = 1 \ldots m \end{array} \tag{7}$$

Though we have removed dependency, formation (7) still poses a significant challenge as it involves non-linear optimisation. Any solution that is locally optimal may not be globally optimum. Conversely, to find a global x^*, we need a "seed" that is very near in order to initiate the search.

We make the observation that tolerance is a very small quantity compared to dimension. Assuming that a nominal dimension $\bar{x}$ (without tolerance) is near optimal, we can then use it as the seed to search for a toleranced dimension x^* which is β away from $\bar{x}$.

Status

We are implementing the software for solving the non-linear optimisation problem (7). To verify our conjecture that $\bar{x}$ is near x^*, we intend to compare β with tolerances suggested by ANSI (on a range of $\bar{x}$).

References

[1] E. M. Mansor. "The Application of Probability to Tolerances Used in Engineering Designs," *Proc. Inst. Mech. Eng.*, Vol. 178, No. 1, pp. 29-51, 1963.

[2] A. M. Hasofer and N. C. Lind. "Exact and Invariant Second-Moment Code Format," *ASCM J. Eng. Mech. Div.*, Vol. 100, No. EM1, pp. 111-121, 1974.

2.2.7 Automatic Determination of Retrieving Paths in a Semantic Data Model

Investigator: Y.C. Lee

Problem Description and Research Objectives

In an environment such as manufacturing, it is inevitable that a great deal of useful information will be stored in many existing databases which, most likely, have different structures and/or use different query languages. It is also true that most of these databases are still being retrieved and updated by their own users. On the other hand, as information management techniques have progressed, more new knowledge and existing experience has been organised in the form of rules and high level operators that are quite different from the traditional formatted data types.

Semantic models for databases [1] [2] [3] [4] [5] that result in more meaningful databases and provide meta-knowledge about the data seem to be good candidates for the integration of various systems. However, methods to exploit this increased information so that the task of the user is simplified have been few in comparison to the number of semantic models in recent years. Accordingly, while being highly expressive, semantic data models also present the user a much more complicated view of data to deal with.

The Proposed Approach

In the first phase of this research, we adopt the Semantic Association Model (SAM[5]) as the integration means and the objective is to equip the model with inferential capabilities. We define the concept of a path with respect to a query in a data model as a structured set of data units (atomic or molecular) necessary for answering the query, and the problem of path determination which involves the automatic determination of the path and its use in answering the query.

Path Determination

Every request references a set of attributes of the associations (data units) of the data model. The values of some of these attributes need to be determined whereas the rest of the attributes may have single values or a range of values or some relationship specified between them. Attributes whose values are required by the query are called unknown attributes and their corresponding associations are called unknown associations. The rest of the attributes referred to in the query are called known attributes and their corresponding associations known associations. These and some other associations that are yet to be determined are involved in answering the query. The network of associations which forms the Semantic Association Model facilitates the process of finding these associations and answering the query. If the semantic model as a network of associations is accessible as a graph, the set of associations obtained in the path determination process is a subnetwork of the database, and is called the *solution tree.*

We propose an implementation strategy for the Semantic Association Model and based on this implementation suggest a solution for the path determination problem. The user view of the path is a graph with the nodes representing the data units and forms a window into the database. The dataflow between the data units required for answering the query involves a traversal of this graph.

Implementation of the Association of SAM

In order to facilitate the formulation of the path determination problem, the associations of SAM must be implemented. A *relational* implementation using the idea of surrogates (E. F. Codd[1]) is found suitable for the associations as well as path determination and the solution is proposed in terms of this implementation.

A ***key of an association*** is one of the characteristics of the association (or as described later, a surrogate attribute), such that there is a functional 1-1, onto mapping from the set of key values to the set of elements belonging to the association. Since no two elements of an association are identical, the key essentially can be used as a functional mapping from the key value set to each of the attribute values of the association elements. A ***surrogate*** is a permanent system assigned pseudo-attribute of an association which functions as a key for that association (could be Aggregation, Generalization, or Interaction). The need for surrogates arises in order to free the user from keeping track of user controlled keys particularly when the concatenation of two keys is required in order to function as the key of some relational table (association). The added advantage of having surrogates for each association is that the inclusion of the association into its parent associations can be done by including the surrogate as an attribute in the parent association with the attribute name being the association name of the subassociation. Property inheritance for generalizations is also simplified by this process.

The Proposed Path Determination Process

The procedure of answering the query is composed of three phases. In the first phase the associations which are involved in answering the query are determined. This is done using the graph that represents the semantic database and the nodes in this graph which correspond to the required associations are determined. This phase also simultaneously finds the path (a set of associations) along which dataflow occurs in the next two phases. The path is not necessarily a sequence of associations but in general is a tree with the node corresponding to the common root as the root of the tree.

The second phase is called the key gathering phase where the solution tree obtained in phase 1 is traversed bottom-up from the nodes representing the known associations up to the common root. This traversal involves the use of the mapping from each attribute set to the key set of the associations on the path, and stops when the common root is reached at which stage there are one or more key sets obtained. Set operations are now performed on these key sets to determine the solution key set.

The third phase is called the solution and it uses the key set available at the common root at the end of phase 2. It traverses the part of the solution tree from the common root to the nodes which correspond to unknown associations (with respect to the query) and the query is answered at the end of this phase.

Phase 1 operates only on the graph representing the semantic network whereas the phases 2 and 3 do so on the graph as well as the relational tables representing the associations relevant to the query.

Status of Research Effort

In the following, each phase of the path determination process is discussed. Please refer more details to [6].

Phase 1: backchaining

The graph representing the semantic network is traversed yielding a subgraph called the solution tree. The nodes of the graph corresponding to the associations whose attributes are referenced in the query are *triggered.* These are the known as well as the unknown associates of the query and are called start nodes. Subsequently, each start node triggers the nodes that are parents of it which in turn trigger parent nodes of them. Only those nodes which are not triggered earlier are triggered at each step. Thus, a front of nodes is formed and this proceeds up the graph. From each start node is formed a subgraph of nodes triggered directly or indirectly by the start node. The process stops when a node is reached which lies on each of the subgraphs associated with the start nodes. This node is called the common root. At this stage there exists a backchained subpath (a subset of the entire path) from each start node to the common root. Thus, if X_{10} is a start node then there exist nodes $X_{11}, X_{12}, \cdots, X_{1n}$ such that $X_{11}, \cdots, X_{1n}$ is a backchained subpath from the start node X_{10} to the common root X_{1n}. Associated with each start node is such a backchained subpath (called *backpath*) and the subgraph which contains all the backpaths with the associations and the arcs associated with them is called the *solution tree.* In general the backward chaining procedure gives the associations required in order to answer the query.

The Interaction association must be treated slightly differently from the others in phases 1 and 2 of path determination. The problem which can arise is the absence of a common root in the process of backchaining. Further, since the same Aggregation association can be a child of two Interaction associations which are independent, it is necessary to trigger both the parent and child (not including the subassociation which triggered it) associations. In some cases the path can become highly complex depending on the semantic network. If more than one path is present then semantically it is more meaningful to deal with the path at the lower level as it demonstrates the link between two relationships for the information retrieval involving the two relationships.

Phase 2: key gathering

In the second phase the solution tree is traversed bottom-up from the known nodes to the common root to obtain the solution key set. During the traversal key sets are formed and set operations may be performed on them.

From each start node which corresponds to a known association, the backpath to the common root is traced. If $X_1, X_2, \cdots, X_k$ is a backpath for a start node to the common root then there is a set of attribute values of the association X_i, which correspond to a set of key attribute values of the association X_{i-1}. The query specifies a set of attribute values of the association X_1 which therefore corresponds to a set of key attribute values of the common root. Thus, each known association has a corresponding key set at the common root, but during the process of traversing the backpaths some set operations need to be performed between the key set corresponding to different start nodes. The points on the solution tree where these operations are performed depend on the query.

If the backpath involves Interactions and if X_1 is the parent of X_2 then the former need not be the parent of the latter. This alters the second phase slightly as it involves the use of the relational table of the Interaction. However, the function of the common root remains the same in the second and the third phase even if Interactions are involved.

The inheritance of common attributes of a Generalization association to the associations which form the Generalization occurs automatically as a consequence of the algorithm. The query which requires such inheritance will refer to some common attribute and some attribute of a subassociation of the Generalization which finally results in the triggering of the common properties association and the subassociation involved. This causes the common properties association to function as the common root for the query.

Phase 3: solution finding

The backpath from the nodes corresponding to unknown associations to the common root is traversed top-down to determine the solution to the query. The query requires the value of some attribute which belongs to the unknown association. If $X_1, X_2, \cdots, X_n$ is the backpath from the unknown association to the common root then an attribute in X_l (no values are referred to here but just the attribute names) which in turn corresponds to some attribute in X_{l+1}. Thus, the attribute required by the query indirectly corresponds to some required attribute in each of the associations $X_1, \cdots, X_n$. The association X_n is, of course, the common root association. Available at the common root is the solution key set at the end of phase 2. Now, a set of key values at X_j corresponds to set of values for the required attribute associated with X_j which in turn corresponds to a set of key values in the association X_{j-1}. Thus starting with the solution key set at the common root the required attribute values at the unknown association are obtained by traversing the backpath from the unknown association to the common root in the reverse direction. Thus the query is answered at the node corresponding to the unknown association.

If the backpath involves Interactions the procedure remains unaltered but the Interaction is traversed from one of its subassociations to the other with the second subassociation effectively functioning as the parent association of the Interaction as far as the algorithm is concerned.

Future Work

We have formulated the problem of path determination and proposed an approach to determining the associations (relations) involved in answering some queries, as well as the direction along which data flow occurs. However, problems remain in performing more complicated queries. Further, the solution to the problem of path determination must be generalized to accommodate complicated data types and high level operators.

Immediate Goal

For some types of queries the entire process can be automatized thus freeing the user of the task and of having detailed knowledge of the semantic data model. However, some query types that involve not only key selections but also general joins still cannot be answered directly and this is the immediate goal of this study. Further, the solution tree in general need not be a simple tree structure (with a root) but could be several fragments of subnetworks of the semantic model. The goal is to automatically combine these fragments including other associations so that the path can be defined in such a case as well. In addition to set operators union and intersection, the use of algebraic operators like join and selection will definitely enhance the capability of the algorithm as it would result in some of these query types which involve relations between arbitrary attributes of different associations being answered.

Final Goal

In addition to such a model, methods to communicate with databases that use different structures and/or different query languages must also be developed. As indicated earlier, it is required that the interpretation of queries and constraints between the semantic model and each individual existing database be established.

An expressive semantic data model with inferential capabilities is clearly a desirable tool for the integration of various databases because a new user will be able to access all databases through the semantic association model using simple and unified queries while the access methods for each individual database remain unchanged. The capabilities of such a system, however, should not be so limited. Its inferential power should be enhanced so as to deal with not only conventional data types but also geometric data types, rules, high level operators and so on.

References

[1] E. F. Codd. "Extending the database relational model to capture more meaning," *ACM Transactions on Database Systems*, Vol. 4, No. 4, pp. 397-434, December 1979.

[2] M. M. Hammer and D. J. McLeod. "Database description with SDM: A semantic database model," *ACM Transactions on Database System*, Vol. 6, No. 3, pp. 351-386, September 1981.

[3] N. Roussopoulos and J. Mylopoulos. "Using semantic networks for database management," *Proc. the International Conference on Very Large Databases*, September 1975.

[4] J. M. Smith and D. C. P. Smith. "Database abstractsion: Aggregation and Generlization," *ACM Transactions on Database Systems*, Vol. 2, No. 2, pp. 105-133, June, 1977.

[5] S. Y. W. Su. "Modeling integrated manufacturing data with SAM*," *Computer*, pp. 34-49, January 1986.

[6] Y.C. Lee and M. Doctor. "Automatic Determination of Retrieving Paths in a Semantic Data Model," submitted to the *IEEE Computer Society Symposium on Office Automation*.

2.3 System Integration Techniques

2.3.1 The Robotics Prototyping System

During the first year of this contract a "vision workbench" was developed. This was a prototyping system that grew out of our work in robot vision, specifically our work in the recognition of partially visible objects [1],[2],[3]. It allows researchers to try out ideas quickly. The workbench is far from complete (the hardware is assembled, but a comprehensive software library has yet to be assembled), however, it has attracted a lot of attention and become a heavily used resource in robotics laboratory. Its wide acceptance has prompted us to increase the scope of the workbench to handle sensory inputs other than vision, and to provide the capability to control a variety of effectors. In keeping with its expanded capability the development system is now referred to as the Robotic Prototyping System (RPS).

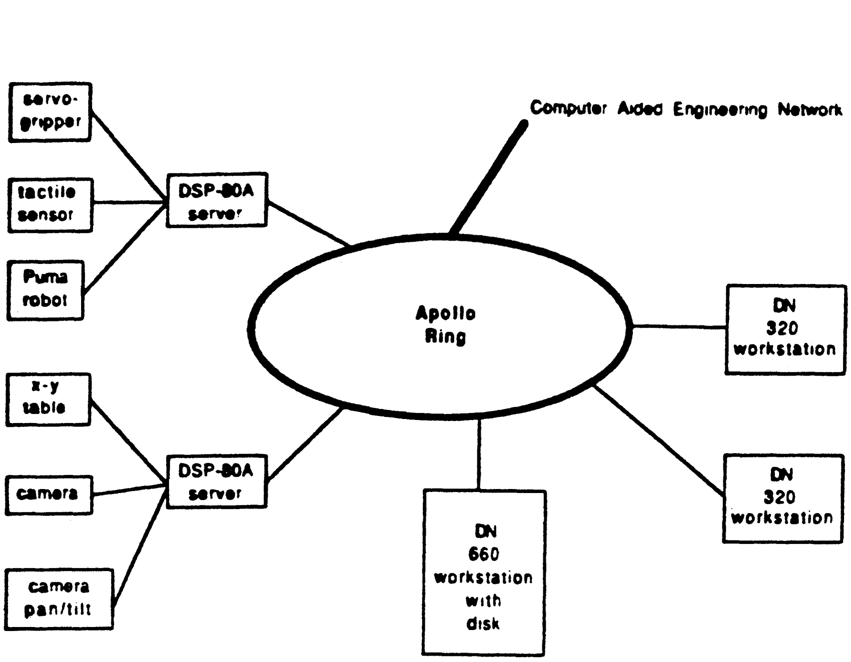


Figure 15: Robot prototyping system.

Figure 15 shows the current hardware configuration of the RPS. The system integrates cameras, a tactile sensor, two PUMA 600 robots with servo grippers, and a X-Y table. Items being added include an additional camera with pan and tilt control and a wrist force sensor. As can be seen from the figure, the system integration is achieved through a network of workstations (we are using the Apollo domain). It is the recent availability of commercial high performance workstations that can be networked and that are capable of displaying high quality graphics that makes the RPS feasible. The network allows the prototyping hardware to be shared among a group of users; eventually it will allow software in the form of reusable software components to be shared. The graphics capability of the workstations allows pictorial representations of the sensor data that, we believe, significantly improves the users ability to conceptualise the results of various experimental steps. However, the intent of this project is broader than simply assembling a state-of-the-art resource for experimental robotics. The long term goal is to establish a research environment in which new and more fruitful modes of research will emerge. Three points that we believe are key ingredients in achieving this are the ability to represent data graphically (although how to represent non-visual data still remains an open question), the capability to acquire sensory data quickly from experiments, and the ability to command the effectors from the same programs that acquire and process the sensory data.

In the case of images for robot vision, it is fairly obvious that the images captured with the cameras are their own best representation, and can be displayed directly on the workstation. Similarly, images that have been processed as part of vision algorithm can also be represented on the workstation directly. So, for instance, the following may occur during the development of an algorithm to extract depth information using stereo views of an object:

> The two images of the object captured by the cameras are displayed initially. Then the processed versions of the images showing noise removal, edge detection, feature extraction, and correlation operations between features in both of the images. This might be followed by annotated images that depict the effects of trying to make correspondences between the features in the two images.

At each phase the windowing capability can be used to display multiple images in various stages of processing in addition to the code that operates on them. This instant access to a representation of the sensory data and the code associated with extracting the "meaning" from that data will, we believe, result in the new modes of research mentioned above. In the case of non-visual sensory data, representing it graphically is an important issue if the above research paradigm is to succeed.

The RPS is currently capable of performing the above stereo vision scenario because the work so far has focussed on vision applications. A more challenging problem is to invent ways to represent non-visual data. Figure 16 shows an example of our initial attempts at this with the Lord tactile sensor. The sensor is organized as a 10 × 16 array of elements. The sensor is capable of indicating, among other things, 4 bits of deflection at each element. Figure 16 shows a plot of the deflection when the robot arm with the sensor grips an object. Clearly there is a strong similarity between the gray levels at the photo-sites of a camera and the deflection levels at each element of the tactile senor. We are currently using this analogy as a guide to exploring representations for the tactile sensor. For example, edges in a visual scene are loci of high gray level gradient; by analogy edges in a "deflection image" are loci of high deflection gradient as may occur when the gripper grasps a block by its corners.

The work we plan for the next year is as follows:

- Addition of more sensors. In particular, a camera with pan and tilt controllable from the workstations. Preliminary work has begun.
- Integration of GEOMOD, a solid modeling program, into the system. This will allow the creation of synthetic images to test out ideas when real scenes are not

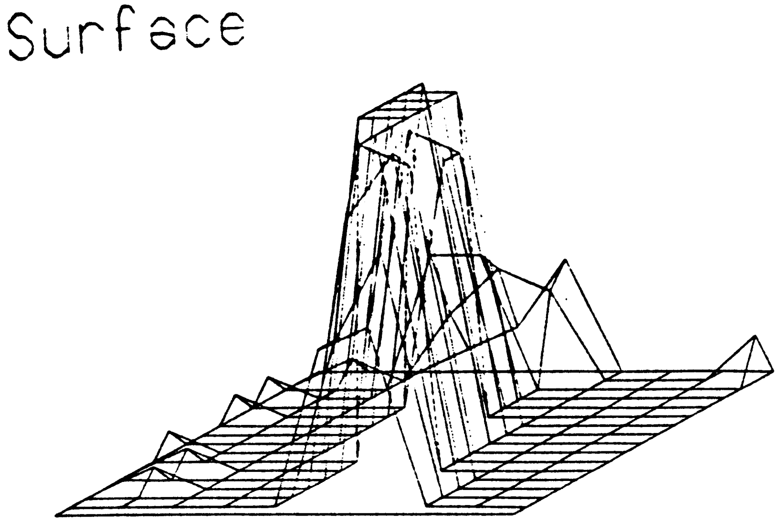


Figure 16: Visual representation of tactile data.

appropriate, as for example if a segmentation algorithm is to be tested on an "impossible" figure (such as an Escher staircase), or for example when a noise free image is required. Preliminary work has begun and GEOMOD has been used to test out some ideas in 3D vision.

- Evaluate a suitable range sensor to integrate into the RPS.
- Add an array processor as a network server so that users can ship low-level image processing work to it to be done quickly. Preliminary work has been done to interface a Mercury array processor.
- Develop a coherent library of optimised algorithms for the array processor.
- Interface our hypercube multiprocessor (see net section) to the RPS network to provide a massively parallel processor resource to the users.
- Continue the collection and development of a coherent set of reusable software components for the RPS. The user community is an important source of software components. We are in the process of defining interface standards so that users can share their software.

2.3.2 Research into Vision Algorithms

A characteristic of much of the work in robotics is the need to process large sets of data. This is particularly true of vision work where single images typically contain $\frac{1}{4}$ megabytes of data from a standard camera with 512 x 512 photosites. If multiple views are required (as in our stereo example above) this can grow further, and if video frame rate processing is required as in the case of dynamic scene analysis (c.f. Ramesh Jain's work) as much as 10 megabytes of raw image data needs to be processed per second. Furthermore, near term developments will give us color images (a factor of 3 increase in data) and finer pixel resolution from the cameras (4096 × 4096 instead of 256 × 256), all of which are going to lead to continued increases in the data sizes.

One solution that has been proposed to deal with the enormous computation requirements of robot vision has been parallel processing. This is not a new idea, but it is a particularly attractive one for low-level robot vision tasks like edge detection, because many of these low-level tasks are simply repeated neighborhood transformations applied to the two dimensional array of image data. Such operations are inherently parallel. Another development that contributes to the attractiveness of parallel processing is the emergence of very high performance 32-bit microcomputers with the functionality and performance of supermini class computers. Assembling large numbers of these to form massively parallel machines is now a possibility.

Our work in this area has focussed on developing algorithms for a massively parallel hypercube multiprocessor, the NCUBE/six. An n-dimensional hypercube or (binary) n-cube computer is a multiprocessor characterized by the presence of $N = 2^n$ processors interconnected as an n-dimensional binary cube. Each processor P forms a node (vertex) of the cube with its own cpu and local main memory. P has direct communication paths to n other processors (its neighbors), which correspond to the edges of the cube that are connected directly to P. There are 2^n distinct n-bit binary addresses or labels that may be assigned to the processors so that each processor's address differs from that of each of its n neighbors in exactly one bit position. Figure 17 illustrates the hypercube topology for $n = 4$, a 4-cube. Our version of the NCUBE multiprocessor is a 6-cube, i.e., it has 64 processors each of which is a 32-bit cpu with 128 K-bytes of memory. The performance of a single node is comparable to that of most state-of-the-art 32-bit machines, which in turn are markedly better than older superminis such as the VAX class machines. The largest system in the series is an NCUBE/ten which is a 10-cube (1024 processors). Our algorithms are designed to scale accordingly.

For some time, it has been known that the hypercube structure has a number of features which make it a useful architecture for parallel computation. For example, meshes

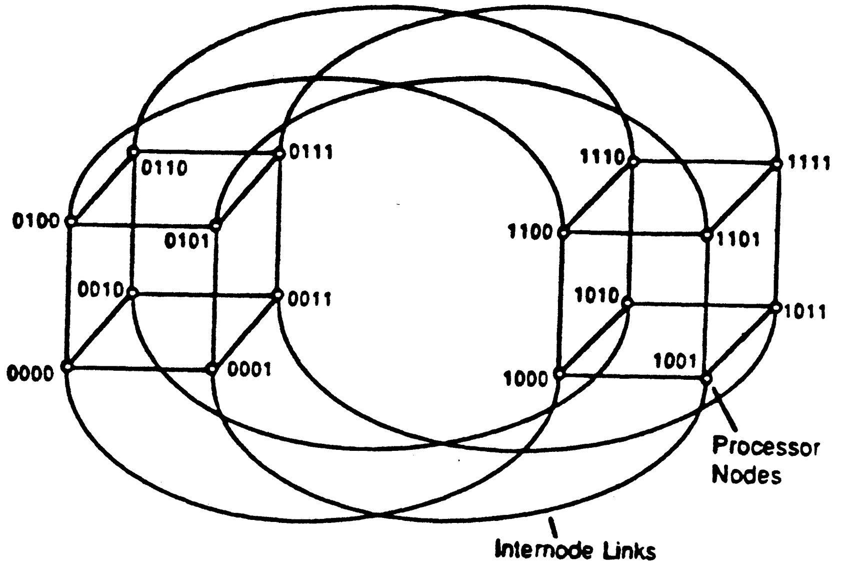


Figure 17: A 4-cube.

of all dimensions and trees can be embedded into a hypercube so that neighboring nodes are mapped to neighbors in the hypercube. The communication structures used in the Fast Fourier Transform and Bitonic Sort algorithm can similarly be embedded into the hypercube. Since a great many scientific applications use mesh, tree, FFT, or sorting interconnection structures, the hypercube is a good candidate for a general-purpose parallel architecture. Even for problems with less regular communication patterns, the fact that the hypercube has a maximal internode distance (the graph diameter) of $n = \log_2 N$ means any two nodes can communicate fairly rapidly. This diameter is larger than the unit diameter of a complete graph K_N, but is achieved with nodes having only degree or fanout of $\log_2 N$, as opposed to the $N-1$ degree of nodes in K_N.

We have benchmarked a number of low-level and mid-level computer vision algorithms on the NCUBE/six the results have been reported in [4], [5], [6]. The results have been encouraging and current work is now focussed on high-level algorithms. The first step in this work is to characterise high-level algorithms for robotics. Our initial attempts are examining generalised branch and bound (B&B) algorithms. A large class of algorithms such as backtracking, searching, and-or search, and rewriting systems can be viewed as B&B, and these are the kinds of algorithms that one associates with "reasoning" about a scene, particularly at the level where multisensor integration is being performed.

Currently the NCUBE is available through a 19.2 K-baud link from the RPS (see above). Next year we plan the following work.

- Wrap up or work on low-level and mid-level computer vision algorithms. An outstanding project is to benchmark the NCUBE with the Abingdon Cross benchmark. This is a synthetically generated image complete with a controlled amount of noise that has been used by Kendall Prestón and others at CMU as a standard benchmark image. There exists a set of performance figures for extracting the medial axis transform (skeleton) of the cross. We would like to compare our results with those of other machines.
- Look at ways of organising B&B algorithms for massively parallel machines like the NCUBE.
- Our long range goal is to make the NCUBE a server on the RPS network: begin work on this project.

References

[1] J.L. Turney, T.N. Mudge and R.A. Vols. "Recognising Partially Hidden Objects," it Proc Soc. of Photo-Optical Engineering, Vol. 521, pp. 108-113, November 1984.

[2] J.L. Turney, T.N. Mudge and R.A. Vols. "Recognising Partially Hidden Objects," *IEEE. Trans. PAMI,* Vol. PAMI-7, No. 4, pp. 410-421, July 1985.

[3] J.L. Turney, T.N. Mudge and R.A. Vols. "Automatic Generation of Salient Features for the Recognition of Partially Occluded Parts," to appear *Robotica,* 1987.

[4] T.N. Mudge. "Vision Algorithms for Hypercube Machines," *IEEE Workshop on Computer Architecture for PAMI Data Management,* November 1985 also to appear *Journal of Parallel and Distributed Computing* 1987.

[5] T.N. Mudge. "The Next Generation of Hypercube Computers," *ARO Workshop on Future Directions in Computer Architecture and Software,* May 1986.

[6] J.P. Hayes, T.N. Mudge, Q.F. Stout, S. Colley and J. Palmer. "Architecture of a Hypercube Supercomputer," *1986 International Conference on Parallel Processing,* August 1986.

2.3.3 Distributed Systems Integration Techniques

Due to the importance of related work on distributed systems integration techniques to this project, we briefly describe that work. Essentially, this work revolves around distributed computing techniques.

Problem Discussion

Distributed computing is an essential part of most large systems today. Until now, however, the principal focus on distributed computing systems has been on their architecture (particularly interconnection mechanisms) and their gross capabilities (usually calculated in some simplistic sense such as multiplying the capabilities of a single processor by the number of processors in the system). One of the most critical problems for the future is the set of programming tools available for such systems.

Software tools for distributed systems must deal with both diverse hardware and the use of existing software written in a wide variety of languages. They must also incorporate the best techniques developed in software engineering over the past two decades and extend these concepts effectively for use in distributed systems[1] It is our hypothesis that extending these concepts to permit distributed execution is a critical step in addressing the distributed computing software problem. Two obvious benefits of so doing are the extension of compile time error checking across machine boundaries and allowing the programmer to use normal language mechanisms for expressing parallel (concurrent) operation without having to invent new application level communication protocols. (Note that computer communication networks do *not* adequately address the applications level protocol issue.) These two advantages alone should greatly improve the problem of developing software for distributed systems.

Example of Using Distributed Program Execution

A particularly interesting application of distributed program execution, arises through various extensions to the "software components" idea prevalent in the Ada community. One major problem in dealing with vendors of (programmable) devices that are attached to computers is that while the purchaser often specifies the hardware capabilities of the device being purchased, he/she rarely is able to specify the software interface. As a consequence the purchaser must spend a great deal of time interpreting vendor documentation (which is often incomplete or erroneous) and trying to wedge the device interface into the rest of the system, i.e., writing a device driver for it which

[1] Among the key concepts are: 1) data encapsulation and hiding, 2) abstract data types, 3) modularization of programs, 4) separate compilation (of both modules and specifications), 5) concurrency mechanisms at the language level, and 6) extensive compile time error checking.

reflects to programmers what the vendor allows them to do (which is often somewhat different from what one needs to do). This problem is much more serious for devices like robots and sensors than it is for more common and standard devices like disk and tape drives.

Through use of a standard language and package concept (ala Ada) in which the specifications for a program module can be compiled separately from the implementations, a powerful alternative is available. The purchaser can write a package specification for the device driver which is compatible with the system the purchaser is developing. Since the specification can be compiled separately from the implementation, the purchase could provide the specification to the vendor and require the vendor to not only meet hardware specifications, but the software specifications as well. After all, the vendor is certainly more familiar with the product than the purchaser. Under the assumption that a truly standard language is used the vendor could develop the implementation to match the specification on whatever hardware he/she happened to have. This is an entirely new way of thinking about purchasing devices to be integrated into a computer system, and depends for its success upon the true standardisation of the language and the separate compilability of module specifications. The idea extends the concept of software components to include devices as well as software. Essentially, the entire device becomes a software component, i.e., an abstract data type [1], from the perspective of the purchaser's computer equipment.

In most cases today, the above idea would involve distributed computing since the body implementing the interface package specification would execute on the purchaser's computer system, while the actual device control would probably execute on a processor attached to the vendor's device. This situation is illustrated in the context of factory automation in Figure 18. Clearly the ability to have distributed execution of concurrent programs would help this style of implementation immensely.

Previous Work and Research Status

The implementation of a system to support distributed program execution for real-time applications requires the solution to a substantial number of problems. We review pertinent work in four areas:

- Problem analysis,
- Timing mechanisms,
- Performance evaluation, and
- Experimental implementations.

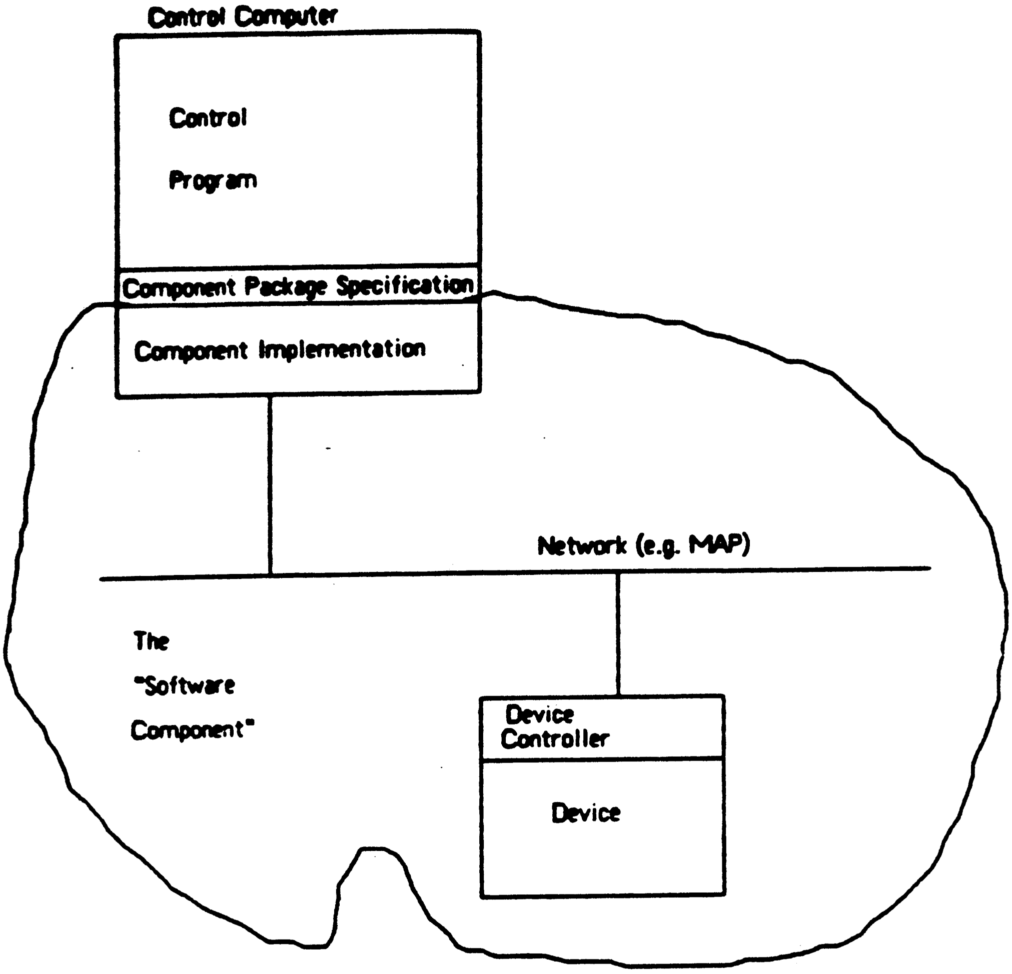


Figure 18: A distributed "software component" view of a component to a manufacturing cell

Problem Analysis: We have studied the basic issues of definition of a distributed program, identified the basic dimensions to the problem, and determined the implications of several alternatives to solving some of the fundamental issues involved. These results are described in detail in [2]. In summary, we have identified the architecture of the memory interconnection, the binding time and the degree of homogeneity as three basic dimensions to the problem of distributed program execution. Changes in any of these dimensions are likely to have significant changes to the implementation strategies developed.

Within the language itself, a specification of what units of the language may be distributed, and how this distribution is to be specified, is crucial. It is shown in [2] that distribution of packages, subprograms, tasks or blocks implies a requirement for access to remote objects in Ada. Further, if subprograms, tasks or blocks are distributed, recursion problems which are inefficient and awkward to implement arise. These problems can be avoided if only library packages and subprograms are distributed.

Major issues arise with respect to the declaration of allocation of objects whose types are defined in units residing on remote processors. It appears that placing objects together with redundant instances of their basic and implicit operations on the machine on which the objects are declared is the best choice, with user defined operations remaining on the processor holding the unit which elaborated the type definition. This interpretation means that user defined operations on objects of remotely defined types will imply remote subprogram calls. [2] also discusses the implications of task termination across machine boundaries and some preliminary conclusions regarding binding time and heterogeneous machine implications.

Our analysis of these issues now forms the basis for a spectrum of future research.

Timing Mechanisms: One of the most fundamental issues for real-time applications in the distributed environment is the management of time across the network. In [3] we have addressed the timing issue at the machine level, and proposed new instruction level timing mechanisms that can considerably simplify the types of timing mechanisms which must be implemented in an Ada run-time system. We also show how these can be used to simplify timing control at both the user and system level in Ada programs. It is also possible to implement the most essential parts of the idea, albeit less efficiently, totally within software. The essential requirement used in formulating these ideas was the use of an (relative) absolute sense of time rather than the interval sense of time implied by the language semantics.

The use of an absolute sense of time is also the basis for maintaining time synchronization across multiple processors. The implications of network timing on the interpretation of the definitions of conditional and time entry calls in Ada were introduced in [4]. [5] explores these issues raise in [4] more completely and proposes an implementable set of interpretations which rigorously define the semantics associated with these calls. This paper also surveys and explores the possible mechanisms for maintaining synchronism among clocks on different processors in the system.

Experimental Implementation: As a basis for experimentally identifying the fundamental problems associated with distributed execution systems, we have built a translation system for a subset of Ada that allows distributed execution of Ada programs, with subprograms, blocks and tasks as units of distribution. The system is based upon using pragmas (SITE) to specify statically the distribution of program elements, and a pre-translator which translates an Ada program with SITE pragmas into a set of Ada programs, one for each computer in the network. Appropriate translations are made for off-machine references by utilising an underlying remote call and mail mechanism. Some of the essential ideas for this translation are described briefly in [4]. As a demonstration of the translator system, an autonomous vehicle with a TV camera mounted on it for vision was implemented in our distributed Ada, with the processing for the vision data being done on one computer and the vehicle control on another. Yet only a single program was written.

On the basis of what we learned from our first experimental translator and our analysis of the problems associated with distributed execution [2] we have begun the development of new translation strategies and a second generation translation system for distributed execution. The new system is based upon distribution library packages and subprograms, and is expected to be in the initial phases of operation by the end of calendar 1986.

Performance Evaluation: A large set of real-time performance benchmarks have been developed to analyze the efficiency of code generated by Ada compilers [6]. They cover a broad range of basic language features and elements of the run-time system. This study also provided insight into proper benchmarking techniques and laid the foundation for future benchmark development. This work has rapidly come to be regarded as the most carefully and thoroughly done that is available. It is the most likely basis for a major performance evaluation development to be started in industry in 1987.

Some additional performance monitoring on an informal level was done while investigating ways to use tasking in finding all solutions to the eight queens problem simultaneously [7]. Two solutions using tasking and allowing for parallelism were examined as well as a sequential solution. There was also discussion on what types of architectures are best suited for the proposed solutions.

Future Directions

We have identified a broad range of areas where research in distributed languages, in general, and Ada, in particular, are important. We describe briefly the directions we believe it important to pursue.

Definitional issues: Although we have already addressed many of the definitional issues, there are two areas remaining, the manner in which the distribution specification is expressed and issues pertaining to heterogeneity.

Implementation Strategies and Translation Techniques: Implementation and translation issues can be divided into three categories: 1) basic cross machine intertask communication mechanism, 2) language feature specific techniques, and 3) compilation environments. All, of course, are impacted by the point in the basic problem space (recall that the dimensions are memory architecture, binding time and degree of homogeneity) for which the translation system is targetted.

Distributed Run-Tim Support: A major area requiring research is that of run-tim support. For embedded systems we expect to identify and implement new operations for the run-time system in order to support distributed execution. Examples might be clock synchronization among processors, inter-process communication an I/O server to coordinate simultaneous I/O requests from several processors, or task termination. Further, methods for propagating exceptions and handling abortions across machines may be profitably placed within a distributed run-time system.

Software Tools: In order to effectively program Ada in a distributed environment, several new software development tools are needed. Principal among these is a distributed debugging facility. Although the tasking function is Ada usually allows for simulation of a distributed system and hence a level of debugging on a single processor, a target level debugger will still be needed. Since the system is real-time, there are many conditions that will be difficult or impossible to simulate on a host system.

References

[1] R.A. Volz, and T.N. Mudge. "Robots are (nothing more than) Abstract Data types," *Proceedings of the Robotics Research Conference: The Next 5 Years and Beyond,* Lehigh PA, August 14-16, 1984.

[2] R.A. Volz, T.N. Mudge, G. Buzzard and P. Krishnana. "Translation and Execution of Distributed Ada Programs: Is It Still Ada?," To appear for the Special Issue on Ada of *IEEE Transactions on Real-Time Systems,* Spring 1987.

[3] R.A. Volz and T.N. Mudge. "Instruction Level Mechanisms for Accurate Real-Time Task Scheduling," accepted by *1986 Real-Time Systems Symposium*, New Orleans, LA, December 1986.

[4] R.A. Volz, T.N. Mudge, A.W. Naylor and J.H. Mayer. "Some Problems in Distributing Real-Time Ada Programs Across Machines," *Ada in use, Proceedings of the 1985 International Ada Conference,* Paris, France, May 1985, pp. 72-84.

[5] R.A. Volz and T.N. Mudge. "Ada Timing in Distributed Systems," accepted for Special Issue *IEEE Transactions Computers on Real-Time Systems,* 1986.

[6] R.M. Clapp, L. Duchesneau, R.A. Volz, T.N. Mudge and T. Schultze. "Toward Real-Time Performance Benchmarks for Ada," *Communications of the ACM,* August 1986.

[7] R.M. Clapp, T.N. Mudge and R.A. Volz. "Solutions to the n Queens Problem Using Tasking in Ada," submitted to *SIGPLAN Notices,* 1986.

3 Publications

3.1 Journal Publications

Control of Mechanical Manipulators

1. Iskanderani, A.I. and N.H. McClamroch, "Signal Estimation for Second-order Vector Difference Equations," *IEEE Trans. on Automatic Control* Vol.AC 30, 1985, pp. 771-773.

2. Shin, K.G. and M.E. Epstein, ."A Unified Method for Evaluating Real-time Computer Controllers and Its Application," *IEEE Trans. on Automatic Control.* Vol. AC-30, No. 4, April 1985.

3. Chalhoub, N.G. and A.G. Ulsoy, "Dynamics Simulation of a Leadscrew Driven Flexible Arm and Controller," *Journal of Dynamics Systems, Measurement and Control*, Vol 108, No. 2, pp. 119-126, June 1986.

4. Gilbert, E.G. and I. Ha, "A Complete Characterization of Decoupling Control Laws for a General Class of Nonlinear Systems, *IEEE Trans. on Automatic Control* Vol. AC-31, pp. 823-830, September 1986.

Vision for Robots

1. Knoll, T.F. and R. Jain, "Recognizing Partially Visible Objects Using Feature Indexed Hypotheses," *IEEE Journal of Robotics and Automation,* Vol. 2, No. 1, pp. 3-13, March 1986.

2. Besl, P. and R. Jain, "Invariant Surface Characteristics for 3-D Object Recognition in Depth Maps," *Computer Vision, Graphics and Image Processing,* Vol. 33, pp. 33-80, 1986.

3. Haynes, Susan and R. Jain, "Event Detection and Correspondence," *Optical Engineering* Vol. 25, No. 3, pp. 387-393, March 1986.

4. Angell, D.K. and E.N. Leith, "Incoherent Spatial Filtering with Grating Interferometers," *Applied Optics,* Vol. 24, No. 18, pp. 2903-2906, September 1985.

5. Angell, D.K. and E.N. Leith, "Generalize Incoherent Optical Processing with Grating Interferometers" *Applied Optics,* Vol 25, No. 4, pp. 499-502, February 1986.

Architecture of Robot-Based Manufacturing Cells

1. Antonelli, C., R.A. Volz and T.N. Mudge, "Decomposition and Simulation of Manufacturing Cells Using Ada," *Simulation,* Vol. 46, No. 4, pp. 141-152, April 1986.

2. Shin, K.G. and H.K. Lee, "Communication Task Approach to the Distributed Realization of an Integrated Multi-Robot System," *IEEE Trans. on Software Engineering,* January 1986.

Planning and Programming

1. Shin, K.G. and N.D. McKay, "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators," *IEEE Trans. on Automatic Control,* Vol. AC-31, No. 6, pp. 491-500, June 1986.

2. Shin, K.G. and N.D. McKay, "Selection of Minimum Time Geometric Paths for Robotic Manipulators," *IEEE Trans. on Automatic Control,* Vol. AC-31, No. 6, pp. 501-511, June 1986.

Integrated Solid-State Sensors for Closed-Loop Robot Control

1. Choi, I.H. and K.D. Wise, "A Silicon Thermopile-Based Infrared Sensing Array for Use in Automated Manufacturing,"" *IEEE Trans. on Electron Devices* Vol. ED-33, No. 1, pp. 72-79, January 1986.

3.2 Conference Presentations

Control of Mechanical Manipulators

1. Shin, K.G. and C. Lee, "Compliant Control of Robotic Manipulators with Resolved Acceleration," *Proc. of 24th Conference on Decision and Control,* Ft. Lauderdale, FL, December 1985.

2. McClamroch, N.H., "Compliance at the End Effector of an Electrohydraulically Controlled Robot," *Proc. of 24th Conference on Decision and Control,* Ft. Lauderdale, FL, December 1985.

3. McClamroch, N.H., "Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems, *IEEE Conference on Robotics and Automation,* San Francisco, April 1986.

4. Gilbert, E.G. and I. Ha, "A Complete Characterization of Decoupling Control Laws for a General Class of Nonlinear Systems," *1985 Conf. on Information Sciences and Systems.* Johns Hopkins University, 1985.

5. Ulsoy, A.G., "Utilization," *Proceedings of the American Control Conference,* Seattle, WA, June 1986.

Vision for Robots

1. Turney, J., T.N. Mudge, and R.A. Volz, "Solving the Bin of Parts Problem," *Proceedings of Vision '86,* Detroit, Michigan, June 1986.

2. Besl, Paul and R. Jain, "Range Image Segmentation Using Symbolic Surface Descriptions," *The Fourth Annual Conference on Intelligent Systems and Machines,* Oakland University, Rochester, MI, April 1986.

3. Knoll, T.F. and R. Jain, "Recognizing Partially Visible Objects Using Feature Indexed Hypothesis," *Conference IEEE Robotics and Automation,* San Francisco, April 1986.

4. Dolezal, R.M., T.N. Mudge, J.L. Turney and R.A. Volz, "Determining the pose of an object," *Proceedings of the Society of Photo-optical Instrumentation Engineering 2nd International Symp. on Optical and Electro-Optical Applied Science and Engineering,* Cannes, France, December 1985.

5. Knoll, T.F., "Recognizing Partially Visible Objects Using Feature Indexed Hypothese," *Conference IEEE Robot and Automation,* San Francisco, April 1986.

Architecture of Robot-Based Manufacturing Cells

1. Humoud, Al-Sadoun, O.A. Olukotun, and T. N. Mudge, "Interconnecting Off-The-Shelf Microprocessors," Proceedings of *National Computer Conference, AFIPS,* Vol. 54, pp. 1758-181, July 1985.

Planning and Programming

1. Gilbert, E.G. and D.W. Johnson, " Minimum Time Robot Path Planning in the Presence of Obstacles," *Proc. of 24th Conference on Decision and Control,* Ft. Lauderdale, FL, December 1985.

2. Gilbert, E.G. and I.J. Ha, "Robust Tracking in Nonlinear Systems and Its Applications to Robotics," *Proc. of 24th Conference on Decision and Control,* Ft. Lauderdale, FL, December 1985.

3. K.B. Irani and D.G. Shin, "Knowledge Representation Using an Extension of A Many-Sorted Language," *First Conference on Artificial Intelligence Applications,* 1985.

4. K.B. Irani and D.G. Shin, "Partitioning of a Relational Database Horizontally Using a Knowledge-Based Approach," *ACM-SICMOD International Conference on Management of Data,* 1985.

5. Volz, R.A., T.N, Mudge, A.W. Naylor and B. Brosgol, "Ada in a Manufacturing Environment," *5th Annual Control Engineering Conference,,* Rosemont, IL, May 1986. *1986 IEEE Int'l Conference on Robotics and Automation,* San Francisco, CA, April 1986.

6. Shin, K.G. and N.D. McKay, "Minimum-Time Trajectory Planning for Industrial Robots with General Torque Constraints," *1986 IEEE Int'l Conference on Robotics and Automation,* San Francisco, CA., April 1986.

7. Shin, K.G. and N.D. McKay, "Automatic Generation of Trajectory Planners for Industrial Robots," *1986 IEEE Int'l Conference on Robotics and Automation,* San Francisco, April 1986.

8. Barber, J., R.A. Volz, R. Desai, R. Rubenfeld, B. Schipper and J. Wolter, "Automatic Two-Fingered Grip Selection,"*IEEE Int'l Conference on Robotics and Automation,* San Francisco, CA, April 1986.

9. Volz, R.A. and A. Naylor, "A Panel Discussion on the NSF Workshop on "Manufacturing Systems Integration," *IEEE Int'l Conference on Robotics and Automation,* San Francisco, CA, April 1986.

10. Volz, R.A., J. Wolter, J. Barber, R. Desai and A. Woo, ' Automatic Determination of Gripping Positions," *NATO Advanced Research Workshop,* Pisa, Italy, September 1986.

3.3 Under Review

Journals

1. Shin, K.G. and M.E. Epstein, "Intertask Communications in an Integrated Multi-robot System," to appear in *IEEE J. of Robotics and Automation,*

2. Shin, K.G. and N.D. McKay, "Robust Trajectory Planning for Robotic Manipulators under Payload Uncertainties," submitted to *IEEE Trans. on Automatic Control.*

3. Shin, K.G. and N.D. McKay, "Automatic Generation of Trajectory Planners for Industrial Robots," to appear *IEEE Journal of Robotics and Automation.*

4. Shin, K.G. and H.K. Lee, "Communication Task Approach to the Distributed Realization of an Integrated Multi-robot System," to appear *Computer System Science and Engineering,*

5. Irani, K.B. and S.I. Yoo, "A Methodology for Solving Problems: Modelling and Generating Heuristics," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence Conference.*

6. Gilbert, E.G. and D.W. Johnson, " Minimum Time Robot Path Planning in the Presence of Obstacles," submitted to *IEEE Control System Magazine.* Ft. Lauderdale, FL, December 1985.

7. Gilbert, E.G., D.W. Johnson and S.S. Keerthi, "A Fast Procedure for Computing the Distance between Complex Objects in Three Space," submitted to *IEEE Int'l Conference on Robotics and Automation* and to *IEEE Journal on Robotics and Automation.*

8. Lee, Y.C. and M. Doctor, "Automatic Determination of Retreiving Paths in a Semantic Data Model," submitted to *IEEE Computer Society Symp. on Office Automation* September 1986.

9. Lee.Y.C. and K.-F. Jea, "PAR: A Representation Scheme for Rotational Parts," submitted to *1987 IEEE Conference on Robotics and Automation.*

10. Mudge, T.N., J. Turney and R.A. Volz, "Automatic Generation of Salient Features for the Recognition of Partially Occluded Parts," to appear *Robotica,* 1987.

11. Mudge, T.N. and T. Abdel-Rahman, "Architectures for Robot Vision,' to appear *Specialized Computer Architectures for Robotics and Automation,*, Ed. J. Graham, Gordon and Breach Inc. 1986.

12. Chalhoub, N.G. and A.G. Ulsoy, "Control of Flexible Robot Arm Experimental and Theoritical Results," accepted for publication *Journal of Dynamics Systems, Measurement and Control.*

13. Chalhoub, N.G. and A.G. Ulsoy, "Dynamic Modeling of a Leadscrew and Its Implication in Robotics," submitted to *Journal of Dynamics Systems, Measurement and Control.*

Conferences

1. Irani, K.B. "A Systematic Approach to Subgoal Ordering and Its Application in Heuristic Problem Solving," *The 3rd IEEE Conference on AI Applications.*

2. Volz, R.A. and T. Mudge, "Instruction Level Mechanisms for Accurate Real-Time Task Schedule," to appear *Real-Time Systems Symposium,* December 1986.

3. Nesse, T. K.G. Shin, and R. Throne, "High-Level Design of a Spray Finishing Robot Controller," submitted to *Int'l Symposium on Industrial Robot 1987.*

4. Shin, K.G. and R. Throne, "Robot Path Planning Using Geodesic and Straight Lie Segment with Voronoi Diagrams," submitted to *1987 IEEE Conference on Robotics and Automation.*

5. Naylor, A., R. Volz, R. Jungclas, P. Bixel, and K. Lloyd, "PROGRESS-A Graphical Robot Programming System," submitted to *1987 IEEE Conference on Robotics and Automation.*

6. Volz, R. and R. Desai, "Contact Formulation - Toward a New Approach to Local Motion in Mechanical Assembly," submitted to *1987 IEEE Conference on Robotics and Automation.*

7. Knoll, T.K. and R. Jain, "Learning to Recognize Objects Using Feature Indexed Hypothesis" submitted to *National Conference on Artificial Intelligence,* August 1986.

8. Besl, P. and R. Jain, "Segmentation through Symbolic Surface Descriptions" submitted to *IEEE Conference on Computer Vision and Pattern Recognition,* 1986.

3.4 Technical Reports

1. Neil D. McKay, "Minimum-Cost Control of Robotic Manipulators with Geometric Path Constraints - A Dissertation," RSD-TR-16-85, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, October 1985.

2. Paul Besl, Kurt Skifstad and Ramesh Jain, "Objective Dimensionality Reduction Using Out-of-Class Covariance," RSD-TR-17-85 Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, November 1985.

3. Shih-Ping Liou, and Ramesh Jain, "Detecting Road Edges Using Hypothesised Vanishing Points," RSD-TR-18-85 Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, November 1985.

4. Suk In Yoo, "A Methodology For Solving Problems in Artificial Intelligence," RSD-TR-20-85, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, December 1985.

5. James Barber, Richard A. Vols, Rajiv Desai, Ronitt Rubenfeld, Brian Schipper and Jan Wolter, "Automatic Two-Fingered Grip Selection," RSD-TR-21-85, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, December 1985.

6. Doñg Guk Shin, "A Extension of a First-Order Language and Its Applications," RSD-TR-22-85, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, December 1985.

7. Heung K. Lee and Kang G. Shin, "Communication Task Approach to the Distributed Realisation of an Integrated Multi-Robot System," RSD-TR-1-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, January 1986.

8. N. Harris McClamroch, "Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems," RSD-TR-2-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, January 1986.

9. Ramesh Jain, Sandra L. Bartlett and Nancy O'Brien, "Motion Stereo Using Ego-Motion Complex Logarithmic Mapping," RSD-TR-3-86, Robot Systems Division,

Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, February 1986.

10. Charles J. Conrad and N. Harris McClamroch, "The Drilling Problem: A Stochastic Modeling and Control Example in Manufacturing," RSD-TR-4-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, February 1986.

11. Paul Besl and Ramesh Jain, "Segmentation Through Symbolic Surface Description," RSD-TR-5-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, February 1986.

12. Russell M. Clapp, Louis Duchesneau, Richard A. Volz, Trevor N. Mudge and Timothy Schultze, "Toward Real-Time Performance Benchmarks for ADA," RSD-TR-6-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, January 1986.

13. Paul G. Ranky, "Program Prospectus for the Simulation, Design and Implementation of a Flexible Assembly and Inspection Cell," RSD-TR-7-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, February 1986.

14. Nabil G. Chalhoub, "Control of a Leadscrew Driven Flexible Robot Arm," RSD-TR-8-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, March 1986.

15. Richard A. Volz, Trevor N. Mudge, Gregory D. Buzzard and Padmanabhan Krishnan, "Translation and Execution of Distributed ADA Programs: Is It Still ADA?" RSD-TR-9-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, April 1986.

16. Paul J. Besl, "Surfaces in Early Range Image Understanding," RSD-TR-10-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, May 1986.

17. Rajeev Agrawal and Ramesh Jain, "An Overview of Tactile Sensing," RSD-TR-11-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, July 1986.

18. Russell M. Clapp, Louis Duchesneau, Richard A. Volz, Trevor N. Mudge and Timothy Schultze, "Toward Real-Time Performance Benchmarks for ADA," RSD-TR-12-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, July 1986.

19. Han-Pang Huang, "Constrained Manipulators and Contact Force Control of Contour Following Problems," RSD-TR-13-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, July 1986.

20. Yung-Chia Lee and Kuen-Fang Jack Jea, "PAR: A New Representation for Rotational Parts," RSD-TR-14-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, July 1986.

21. Charles James Conrad, "Stochastic Modeling and Control of Some Problems in Manufacturing," RSD-TR-15-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, July 1986.

22. Jerry L. Turney, "Recognition of Partially Occluded Parts," RSD-TR-16-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, August 1986.

23. Richard A. Volz and Arch W. Naylor, "Workshop on Manufacturing Systems Integration," RSD-TR-17-86, Robot Systems Division, Center For Robotics and Integrated Manufacturing, University of Michigan, Ann Arbor, Michigan, June 1986.

4 Personnel

4.1 Faculty

There are many ways in which a faculty member or student may be supported by a contract of this type. The names listed on the form DD1473 include only those faculty who received direct salary support from the contract. However, in addition, some faculty received support indirectly through support of their graduate or post doctoral students. This section lists faculty who have either directly received support for a portion of their salaries or received support for one or more of their graduate students.

Professors

Daniel E. Atkins	– Electrical Engineering and Computer Science
Keki B. Irani	– Electrical Engineering and Computer Science
Ramesh Jain	– Electrical Engineering and Computer Science
Arch W. Naylor	– Electrical Engineering and Computer Science
Richard A. Volz	– Electrical Engineering and Computer Science

Associate Professors

Trevor N. Mudge	– Electrical Engineering and Computer Science
Kang G. Shin	– Electrical Engineering and Computer Science
Anthony C. Woo	– Industrial and Operations Engineering

Assistant Professors

Y. C. Lee	– Electrical Engineering and Computer Science
T. Weymouth	– Electrical Engineering and Computer Science

4.2 Students

The number of students who have received support during the first year of this contract exceeds the number of "standard" student appointments indicated in the proposal. There are two primary reasons for this. First, and most importantly, we were able to leverage the student funding to increase the student involvement. In a number of cases we were able to obtain partial fellowship support for the student from internal funds, thus reducing the cost per student and increasing the number of students we could involve in the project. Similarly some students had partial outside fellowship support and only needed supplementary support to be able to participate in the project. The involvement of students with fellowship support in the project is deemed an acceptable activity since the work being performed is generally part of the student's dissertation work. Secondly, through natural attrition, graduation, and staff changeover, some replacement of the students supported occurred during the year. The students listed below, then, include all of the students who have received support during the contract year.

T. S. Abdel-Rahman	EECS[1] Ph.D. expected 1986-87
Robert Dolesal	EECS Ph.D. 1987-88
U. Fayyad	EECS Ph.D. expected 1990
Susan Haynes	EECS Ph.D. expected 1986-87
F. Jiang	ECE Ph.D. expect 1988-89
Dan Johnson	AE[2] Ph.D. expected 1985-86
Suntaeg Jun	EECS Ph.D. expected 1987-88
H-S Kim	CICE Ph.D expected 1986-87
W.J. Lee	IOE Ph.D.
Yi Lu	CICE Ph.D. expected 1987-88
O. Oyenkunle	EECS Ph.D. expected 1988-89
Brian Schipper	M.S. EECS expected 1986
T. Sripradisvarakal	CICE Ph.D. expected 1987-88
Shao Lejun	CICE Ph.D. expected 1988
Jing Xiao	CICE Ph.D. expected 1988-89

[1] Electrical Engineering and Computer Science

[1] Aerospace Engineering

4.3 Degrees Awarded

Paul Besl	Ph.D. in EECS
Nabil Chalhoub	Ph.D. in ME&AM
Il Hyun Choi	Ph.D. in EECS
Kukjin Chun	Ph.D. in EECS
H.P Huang	Ph.D. in AE
Brian Schipper	M.S. in EECS

4.4 Graduate Student Placement

1982 - 1983

Gary Swanson, Ph.D.	Lincoln Labs
Henry Chu, M.S.	U of M Ph.D. Program
Paul Eichel, M.S.	U of M Ph.D. Program
Stacy Leon, M.S.	U of M Ph.D. Program
Mike Steigerwald, M.S.	U of M Ph.D. Program
Jan Wolter, M.S.	U of M Ph.D. Program
Brian Dent, M.S.	ADAPCO, Melville, New York
Myrung Chung Ph.D.	Korean Institute of Technology
Dean Askounis M.S.	FMC Robots, Troy, Michigan

1983 - 1984

Thomas Wielenga, Ph.D.	M.D.I., Ann Arbor, Michigan
Joseph Pincu, M.S.	U. of M. Ph.D. Program
Daniel Barash, M.S.	Harris Corporation, Washington, D.C.
Paul Firehammer, M.S.	unknown
Nirwan Ansari, M.S.	U of M Ph.D. Program
David Gal, M.S.	R.O.L.M. in California
Mark Epstein, M.S.	IBM, Boca Raton, Florida
Neil Nathason, M.S.	IBM, Toledo, Ohio
Chi-Chan Tsang, M.S.	Computerised Office Systems, Ann Arbor
Glen Healy, B.S.	Stanford Graduate Program

Elesh Shah, M.S.	unknown

1984 - 1985

D. Beard, Ph.D.	University of North Carolina
B. Lee, Ph.D.	Purdue University
N. McKay, Ph.D.	General Motors Research Laboratory
J. Ha, Ph.D.	General Motors Research Laboratory
P. Eichel, Ph.D.	Sandia Laboratory
J. Lah, Ph.D.	Intel Corporation
S. Yoo, Ph.D.	Seoul University
D. Shin, Ph.D.	unknown
P. Bixel, M.S.	General Electric
K. Lloyd, M.S.	Bell Laboratories
R. Rubinfeld, B.S.	University of California, Berkeley

1985 - 86

Paul Besl, Ph.D.	General Motors Tech Center
Nabil Chalhoub, Ph.D.	University of Reno
Il Hyun Choi, Ph.D.	IBM - Virginia
Kukjin Chun, Ph.D.	University of Washington
H.P. Huang, Ph.D.	Nationa Taiwan University
Brian Schipper, M.S.E.	unknown

4.5 Permanent Staff

Research Engineers

Al Dobryden
Jerry Turney

Systems Programmers

Chris Born
Ron Theriault

Technician

Rob Giles

Administrative Assistant

Jerry L. Jackson

Secretaries

Dolores Bolsenga
Marianne Moylan
Rosalind Perry
Mary Ann Pruder

5 Coupling Activities

An important aspect of our program is the development of strong coupling activities both within and without the program. Four categories of interaction are being pursued:

- interaction among people participating in the project
- interactions with other university scientific groups
- interactions with industry
- interactions with the Air Force.

The following sections describe the coupling activities which have taken place during the past year.

5.1 Intra Project Interactions

Frequent seminars held with both internal and external speakers provided a means of keeping everyone generally aware of work both within our project and at other institutions around the country. Indeed, at times the frequency of seminars was so high that it was difficult for any one person to attend all of interest. The joint seminar series sponsored by CRIM and the Industrial Technology Institute enabled us to bring truly outstanding speakers from both industry and academia to Michigan. See attachment A.

There are regular technical discussion meetings among members of several of the research groups occurring on a weekly basis. Membership in more than one group is common, particularly among the faculty concerned. Also, most of the faculty are members of several doctoral committees on topics outside of their own immediate research activities. This has proven to a useful form of information diffusion.

5.2 University Interactions

Interactions with people from other universities have taken place primarily through seminars, both those given at the University of Michigan, and those given by Michigan people at other universities.

Outside Seminar Speakers

Among those presenting seminars at Michigan are:

- David Dorenfeld – University of California
  "Topics in Acoustic Emission"

- Inyoung Ham – University of Pennsylvania/Pennsylvania State
  "Computer Automated Process Planning"

- George Tlusty – University of Florida, Gainsville
  "Dynamics of High Speed Milling and Milling Cutter Sensing"

- Jane Ammons – Georgia Institute of Technology
  "Problems in Workstation Configuration for Electronic Assembly Systems"

- Jeffrey A.G. Knight – University of Knottingham
  "Ultrasonic Sensing and CAD for Off-Line Robot Programming"

- Ken-ichi Kanatai – University of Maryland
  "Shape from Texture: General Principals"

- Alberto Segree – University of Illinois
  "Explanation - Based Manipulator Learning Acquired Robotic Manufacturing Schemata"

- Mark Fox – Carnegie Mellon University
  "Experiences in Applying Artificial Intelligence to Manufacturing"

- Shing H. Lee – University of California at San Diego
  "Optical Pattern Recognition"

A copy of the seminar schedule for the Fall term is attached in Appendix A.

Seminars at other Universities given by Michigan faculty

- Professor K.G. Shin – Princeton and Yale (Programming Language and control)

- Professor N.H. McClamroch – University of Toledo, Notre Dame and Michigan State Univesity (Manipulator Control)

- Professor R. Jain – University of Illinois (Computer Vision)

- Professor R.A. Volz - Gao Weibing, Vice Dean Beijing Institute of Aeronautics and Astronautics and Direction, Laboratory of Dynamic Systems and Control - discussion
- Professor Y. Koren – University of California, Los Angeles, Columbia University, Rensselaer Polytechnical Institute and Lehigh University (CNC and Intelligent Manufacturing)

5.3 Interactions with Industry

Interactions with industry have taken place in three forms, seminars presented to industry, seminars presented at Michigan and private technical discussions.

Workshop

Professors A.W. Naylor and R.A. Volz organized and hosted a National Science Foundation sponsored workshop on Manufacturing Systems Integration. Forty experts from both academics and industry, worldwide, participated. The participants identified key research topics critical to the success of future integrated manufacturing systems. The results are available in RSD report RSD-TR-17-86.

Seminars and Briefings for Industry

Two industrial Affiliates reviews were held during the year. Each was a one day technical conference presenting recent research results to the affiliate members. Appendix B contains the current membership list in the Affiliates.

Major technical briefings were held for the following firms:

Silicon Graphics, Inc.
Hewlett-Packard
Buick-Olds-Cadillac (General Motors)
3M Corporation
NASA
General Motors (CPC Division)
General Motors (Trilby Project)
CAMi
SYNDECO, subsidiary of Detroit Edison
Ford Motor Company
Martin Marietta
Texas Instrument
Xerox Corporation
Chrysler Corporation
Vickers
Applied Intelligent Systems
Autoflex
IBM

Seminars given by Michigan faculty for the following industries:

- Professor K.G. Shin — Schlumberger Research Laboratory
- Professor K.G. Shin — Ford Motor Scientific Research Laboratory
- Professor T. Weymouth — General Motors Research Institute
- Professor T. Weymouth — Engineering Summer Conferences, Computer Vision and Image Processing
- Professor R. Jain — Engineering Summer Conferences, Computer Vision and Image Processing
- Paul Ranky -TV Classes and seminar for Electronic Data Systems
- Professors R.A. Volz, T.N. Mudge and A.W. Naylor — Ada & Software Engineering, course for General Dynamics
- Professor R.A. Volz — General Dynamics
- Professor N.H. McClamroch — General Motors Research Institute
- Professor Y. Koren - General Electric corporation

Seminars at Michigan by Industrial People

- Daniel Juliette, General Motors
  "Saturn Manufacturing Concepts"
- R. Hocken, National Bureau of Standards
  "Implementation of FMS and Integration of Inspection in the System"
- L. Burns, General Motors Technology Center
  "Research Opportunities in Manufacturing"
- Larry Sullivan, Suppliers Quality Assured, Inc.
  Improved Design Using Taguchi Experimental Methods
- Kicha Ganapathy, AT&T Bell Laboratory
  "Real Time Motion Tracking Using a Single Camera

Private technical discussions with industry

- Professors L. Conway, E. Gilbert and R.A. Volz met with representatives from Martin Marietta.
- Professors A.W. Naylor, T.N. Mudge and R.A. Volz meet regularly with representative from General Dynamics and General Motors.
- Professor R.A. Volz and A.C. Woo meet regularly with representatives from IBM
- Professor R. Jain meets regularly with representatives from General Motors Research Laboratory.
- Professor T.N. Mudge meets regularly with representatives from Intel, BICOM, N-Cube and Verdix.
- Professor R.A. Volz met with representatives from Westinghouse.
- Professor A.C. Woo met regularly with representatives from Ford Motor Company.
- S. Haynes met with representative from Grumman.
- J. Turney met with representatives of Whirlpool.

5.4 Government Interaction

Meetings with the following DoD and government agencies have taken place during the past year.

- Professors R. Jain, and T. Weymouth – attended a program review and participated in robotics courses at NASA Ames.
- Professor R.A. Volz presented and participated in robotics courses at NASA Ames.
- Professors M. Walker and R.A. Volz – attended Consortium for Space Automation and Robotics meetings sponsored by the Universities Space Research Association (Professor Walker –2 meetings; Professor Volz – 1 meeting)
- Professor R.A. Volz – attends the NASA Automation and Robotics Panel meetings.

- Professor R.A. Volz – attends NASA Aerospace Safety Advisory Panel meetings (approximately monthly).
- Professor N.H. McClamroch attended a NASA Workshop at Ames, "Robotics in Space".
- An annual AFOSR program review was held.
- Air Force planning meetings attended by Professor R. Jain at Dayton, Ohio
- Professor R.A. Volz – was a member of the program committee for a NATO Workshop on Robot Programming Languages
- Professors L. Conway, G. Olson and R.A. Volz presented an overview of robotics and artificial intelligence to NASA headquarters.
- Professor R.A. Volz – participated in NBS AMRF program review

Faculty and Students Giving Seminars within the Uuniversity

- Y. Koren
  "Research Topics in Robotics"
- B. Schunck
  "The Nature and Purpose of Vision Science and Engineering Examples of Future Directions"

  Thomas Knoll – Computer Vision Research Seminar
  "Recognizing Partially Visible Objects Using Feature Indexed Hypotheses" (2 sessions)
- J.L. Turney – Computer Vision Research Seminar
  "Partially Occluded Parts Recognition"
- N.H. McClamroch – Control Seminar
  "Dynamics and Control of Constrained Manipulators"
- J. Birge
  "Real Time Adaptive Scheduling in Flexible Manufacturing Systems"
- S. Bartlett – Computer Vision Research Seminar
  "Motion Stereo Using Ego-Motion Complex Logrithmic Mapping"

- **Shih-Ping Liou – Computer Vision Research Seminar**
  "Road Following Using Vanishing Points"

- **P. Ranky – Business School Seminar**
  Computer Integrated Manufacturing Seminar

- **P. Ranky – Industrial Operations Engineering Seminar**
  Flexible Manufacturing Systems

6 Distribution List

University and Research Institutes

Professor Thomas Binford
Computer Science Dept.
Stanford University
Stanford, CA 94305

Professor Robert McGhee
Dept. of Electrical Engineering
Ohio State University
2015 Neil Avenue
Columbus, OH 43210

Professor Raj Reddy
Computer Science Dept. & Robotics
Carnegie Mellon University
Schenley Park
Pittsburgh, PA 15213

Professor Michael Brady
Cambridge
England

Professor Richard Paul
Dept. of Computer Information
and Sciences
University of Pennsylvania
N60 Town Bldg.
Philadelphia, PA 19104

Professor Roger Nagel
Lehigh University
Bethlehem, PA 18015

Professor Georges Saridis
ECSE
Rensselaer Polytechnical Institute
Troy, NY 12181

Professor Delbert Tesar
Dept. of Mechanical Engineering
University of Texas
Austin, Texas

Dr. David Whitney
Draper Laboratories
555 Technology Square
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. David Nitzen
Director of Robotics Research
Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, CA 94025

Dr. Robert Bolles
Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, CA 94025

Dr. William Brown
ERIM
P.O. Box 8618
Ann Arbor, MI 48107

Dr. Azriel Rosenfeld
Research - Room 4115C
Computer & Space Sciences Bldg.
University of Maryland
College Park, MD 20742

Dr. Geoffrey Boothroyd
Mechanical Engineering
University of Massachusetts
Amhurst, MA 01003

Dr. Jerome Smith
Industrial Technology Institute
Beal Avenue
Ann Arbor, MI 48109

Government

Dr. James Albus
Room A127 Bldg 220
National Bureau of Standards
Washington D.C. 20234

Dr. Howard Moraff
Program Director
Automation and System Integration
Division of Dsign, Manufacturing
and Computer Engineering
National Science Foundation
1800 G. Street N.W.
Washington D.C. 20550

Mr. Norman Caplan
Deputy Division Director
Division of Electrical, Computer &
Systems Engineering
NSF
1800 G. Street N.W.
Washington D.C. 20550

Mr. Bernie Chern
Division of Electrical, Computer &
System Engineering
Room 1101
NSF
Washington D.C. 20550

Captain Robert Delyser
DFEE
U.S. Air Force Academy
Colorado 80840

Dr. James Gault
U.S. Army Research Development
Standardization Group
United Kingdom
P.O. Box 65
FPO New York 09510

Dr. Paul B. Schneck
Head
Dept. of Navy
Information Sciences Division
Office of Naval Research
800 N. Quincy
Arlington, VA 22217

Dr. Vincent Russo
Wright Patterson Air Force Base
AFWAL/MLTC
Wright Patterson, OH 45433

Mr. Thomas Lagnese
Project Director
Computer Integrated
Manufacturing Branch
Department of Air Force
Air Force Wright Aeronautical Lab
Wright Patterson Air Force Base,
Ohio 45433

Mr. Michael Hitchcock
Computer Integrated
Manufacturing Branch
Department of Air Force
Air Force Wright Aeronautical Lab
Wright Patterson Air Force Base,
Ohio 45433

Mr. William M. Spurgeon
University of Michigan at Dearborn
Dearborn, Michigan

Dr. Nam Suh
NSF
1800 G. Street N.W.
Washington, DC 20550

Mr. G. Ronald Green
Electronics Division
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709

Mr. Ted J. Brandewie
Computer Integrated
Manufacturing Branch
Department of Air Force
Air Force Wright Aeronautical Lab
Wright Patterson Air Force Base,
Ohio 45433

Mr. Daniel Herman
NASA Headquarters
Code S
Washington DC, 20546

Major Joe Hager
USAF, AFSC
Air Force Office of Scientific Research
Building 410,
Bolling AFB, D.C. 20332

Industry

Mr. Bertil Thorvaldsson
ASEA
Industrial Robot Division
16250 W. Glendale
New Berlin, WI 53151

Mr. Charles Scott
General Dynamics
Land Systems Division
P.O. Box 1901
Warren, MI 48090

Dr. Steve Holland
Mr. Frank DiPietro
Mr. Gerald Elson
Mr. Gabe Tiberio
Mr. Richard Beecher
General Motors
General Motors Research Laboratories
Warren, MI 48090

Mr. Pete Matheson
Mr. Emil Sarpa
Mr. Anthony Anderson
Intel Corporation
26500 Northwestern Highway
Southfield, MI 48075

Mr. J.L. Escover
Mr. Robert Casey
Lockheed
Missiles and Space Company, Inc.
P.O. Box 1700
Austin, TX 78760

Mr. Dwight Carlson
Perceptron
23920 Freeway Park Drive
Farmington Hills, MI 48024

Mr. Clifford T. Anglewicz
Volvo of America Corporation
Volvo Automated Systems Division
40712 Brentwood Drive
Sterling Heights, MI 48078

Dr. Gale Cutler
Whirlpool
Monte Road
Benton Harbor, MI 49022

Dr. John Klein
IBM
1798 NW 40th Street
Boca Raton, FL 33432

Mr. Bruce Haupt
IBM
951 N.W. 51st.
Boca Raton, FL 33432

Dr. Michael Wesley
IBM
T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Mr. William Hollenback
IBM
Neighborhood Rd.
Kingston, NY 12401

Mr. James Lowrie
Advanced Automation
Technology Section
Martin Marietta Aerospace
P.O. Box 179
Denver, CO 80209

Mr. Donald E. Waters
Program Manager
Technical Strategy Center
Honeywell, Inc.
1700 W. Highway 36
St. Paul, MN 55113

ATTACHMENT A

Dwight Stevenson, Ph.D.,
"The Economics and Practicality of Transferring Technology to Off-Campus Engineers."
Director of the Instructional Television Systems, College of Engineering, University of Michigan

Sep 25

Harold Dewhurst, Ph.D.,
"Electronic Communications. Technology in Industry/University Relations."
Director of Scientific Affairs (retired), Owens-Corning Fiberglas

Oct 9

Sirus Chitsaz,
"The CSPC Research Agenda and Interactions with Industry."
Director, Communications and Signal Processing Center, Department of Electrical & Computer Engineering, North Carolina State University

Oct 23

The Center for Research on Integrated Manufacturing (CRIM) of The University of Michigan, based in the U-M College of Engineering, [illegible]

Christopher LeMaistre, Ph.D.
"CIM, Composites and the Center for Industrial Innovations."
Director, Center for Industrial Innovations, Rensselaer Polytechnic Institute

Nov 6

Jack Russell, Ph.D.,
"The Technology Deployment Service of the State of Michigan."
Acting Director, Innovation & Technology Services, Director, Technology Deployment Services

Nov 20

George H. Kuper,
"The Proposed National Center for Manufacturing Sciences."
Executive Director, Manufacturing Studies Board, National Research Council, National Academy of Sciences

Dec 4

Industry/University Interactions

Fall 1986 Seminar Series

ATTACHMENT B

ROBOTICS INDUSTRIAL AFFILIATES MEMBERS

October 9, 1986

ASEA Robotics, Inc.
Industrial Robot Division
16250 W. Glendale
New Berlin, Wisconsin 53151
Company Contact:
Bertil Thorvaldsson
Telephone: (414)785-3482

Digital Equipment Corp.
APO-1/G2
100 Minuteman Road
Andover, MA 01810
Company Contact:
Dom Zambuto
Telephone: (617) 689-1698

General Dynamics
Land Systems Division
P.O. Box 1901
Warren, Michigan 48090
Company Contact:
Charles Scott
Telephone: (313) 362-8096
1902 Northwood
Troy, MI 48084

General Motors

General Motors Research Laboratories
Warren, Michigan 48090-9057
Company Contact:
Steven W. Holland
Asst. Head
Computer Science Department
Telephone: (313) 575-3203

GMFanuc Robotics Corporation

5600 New King Street
Troy, Michigan 48098
Company Contact
Chia Day
(313) 641-4185

Grumman Aerospace Corporation

Aircraft Systems Division
Mail Station C04/005
Bethpage, New York 11714
Company Contact
Donald J. Heitzmann
(516) 575-7122

Intel Corporation

7071 Orchard Lake Road
Suite 100
West Bloomfield, Michigan 48033
Company Contact:
Pam Alegnani
Telephone: (313) 851-8096

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051
Company Contact:
Mr. Emil J. Sarpa
Corporate Manager Academic Relations

Lockheed
Missiles and Space Company, Inc.
Department T3-35
Building 30-D
2124 East St. Elmo Street
Austin, Texas 78744
Mail to: P.O. Box 1700
Austin, Texas 78760
Company Contact:
J. L. Escover
Manufacturing Engineering Manager
Telephone: (512) 362-7307

Lockheed
Missiles and Space Company, Inc.
Dept. 5301
Bldg. 580
1111 Lockheed Way
Sunnyvale, California 94086
Company Contact:
Mr. S. C. DeBrock
(University Liason)
Telephone: (408) 743-1571

Perceptron
23855 Research Drive
Farmington Hills, Michigan 48024
Company Contact:
Jeremy Salinger
President
Telephone: (313) 478-7710

Westinghouse Corporation
Artificial Intelligence
335 E. Expo-Mart
105 Mall Blvd.
Monroeville, PA 15146
Company Contact
Edward P. Liscio
Telephone: (412) 374-7266

Whirlpool
The Elisah Gray II
Research and Engineering Center
Monte Road
Benton Harbor, Michigan 49022
Company Contact:
Dr. Gale Cutler
Telephone: (616) 926-5215