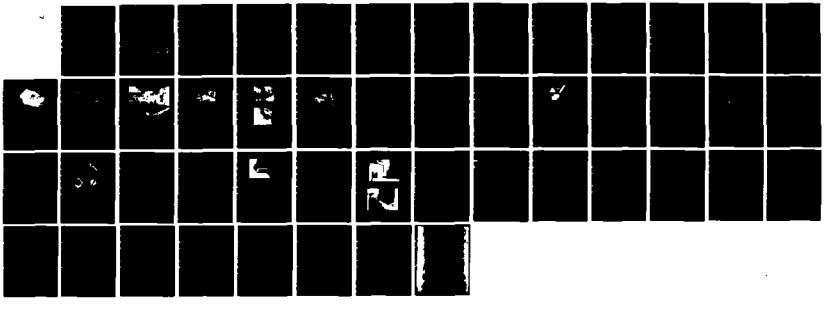
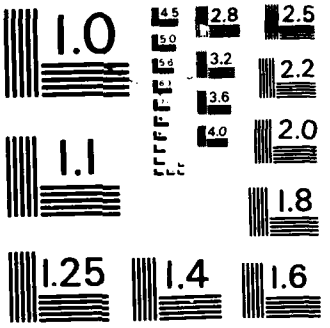
AD-A142 488 ARTIFICIAL INEELLIGENCE AND ROBOTICS(U) MASSACHUSETTS 1/1
INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB
M BRADY FEB 84 AI-M-756 N00014-75-C-0643

UNCLASSIFIED F/G 6/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AIM 756	2. GOVT ACCESSION NO.: AD-A142 488
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Artificial Intelligence and Robotics	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Michael Brady	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0643, N00014-80-C-0505
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, Massachusetts 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Blvd, Arlington, Virginia 22209	12. REPORT DATE: February, 1984
	13. NUMBER OF PAGES: 43
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, Virginia 22217	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC ELECTE JUN 27 1984 S D A

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Robotics
Artificial Intelligence

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Since Robotics is the field concerned with the connection of perception to action, Artificial Intelligence must have a central role in Robotics if the connection is to be intelligent. Artificial Intelligence addresses the crucial questions of: what knowledge is required in any aspect of thinking; how that knowledge should be represented; and how that knowledge should be used. Robotics challenges AI by forcing it to deal with real objects in the real world. Techniques and representations developed for purely cognitive problems, often

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

84 06 20 078

AD-A142 488

DTIC FILE COPY

Block 20 cont.

in toy domains, do not necessarily extend to meet the challenge. Robots combine mechanical effectors, sensors, and computers. AI has made significant contributions to each component. We review AI contributions to perception and object oriented reasoning. Object-oriented reasoning includes reasoning about space, path-planning, uncertainty, and compliance. We conclude with three examples that illustrate the kinds of reasoning or problem solving abilities we would like to endow robots with and that we believe are worthy goals of both Robotics and Artificial Intelligence, being within reach of both.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

DTIC COPY INSPECTED

A. I. Memo 756 February, 1984

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist Avail and/or Special
A-1

ArtificialIntelligence

and

Robotics

Michael Brady

Abstract. Since Robotics is the field concerned with the connection of perception to action, Artificial Intelligence must have a central role in Robotics if the connection is to be *intelligent*. Artificial Intelligence addresses the crucial questions of: what knowledge is required in any aspect of thinking; how that knowledge should be represented; and how that knowledge should be used. Robotics challenges AI by forcing it to deal with real objects in the real world. Techniques and representations developed for purely cognitive problems, often in toy domains, do not necessarily extend to meet the challenge. Robots combine mechanical effectors, sensors, and computers. AI has made significant contributions to each component. We review AI contributions to perception and object oriented reasoning. Object-oriented reasoning includes reasoning about space, path-planning, uncertainty, and compliance. We conclude with three examples that illustrate the kinds of reasoning or problem solving abilities we would like to endow robots with and that we believe are worthy goals of both Robotics and Artificial Intelligence, being within reach of both.

Acknowledgements. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's Artificial Intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643, the Office of Naval Research under contract number N0014-80-C-0505, and the System Development Foundation.

© Massachusetts Institute of Technology 1984

1. Roboticsand ArtificialIntelligence

Robotics is the *intelligent connection of perception to action*. The key words in that sentence are "intelligent" and concomitant "perception". Normally Robotics is thought of as simply the connection of sensing to action using computers. The typical sensing modalities of current robots include vision, force and tactile sensing, as well as proprioceptive sensing of the robot's internal state. The capacity for action is provided by arms, grippers, wheels, and, occasionally, legs.

The software of modern, commercially available, robot systems such as the IBM 7565 [Taylor, Summers, and Meyer 1982], the Unimation PUMA [Val 1980, Shimano et. al. 1984], and the Automatix cybervision [Franklin and VanderBrug 1982, Villers 1982] includes a wide variety of functions: it performs trajectory calculation and kinematic translation, interprets sense data, executes adaptive control through conditional execution and real time monitors, interfaces to databases of geometric models, and supports program development. It does some of these tasks quite well, particularly those that pertain to Computer Science; it does others quite poorly, particularly perception, object modelling, and spatial reasoning.

The *intelligent* connection of perception to action replaces sensing by perception, and software by intelligent software. Perception differs from sensing or classification in that it implies the construction of representations that are the basis for recognition, reasoning and action. Intelligent software addresses issues such as: spatial reasoning, dealing with uncertainty, geometric reasoning, compliance, and learning. Intelligence, including the ability to reason and learn about objects and manufacturing processes, holds the key to more versatile robots.

Insofar as Robotics is the intelligent connection of perception to action, Artificial Intelligence (AI) is the challenge for Robotics. On the other hand, however, we shall argue that Robotics severely challenges Artificial Intelligence (AI) by forcing it to deal with real objects in the real world. Techniques and representations developed for purely cognitive problems often do not extend to meet the challenge.

First, we discuss the need for intelligent robots and we show why Robotics poses severe challenges for Artificial Intelligence. Then we consider what is required for robots to act on their environment. This is the domain of kinematics and dynamics, control, innovative robot arms, multi-fingered hands, and mobile robots. In section 5, we turn attention to intelligent software, focussing upon spatial reasoning, dealing with uncertainty, geometric reasoning, and learning. In section 6, we discuss robot perception. Finally, in Section 7, we present some examples of reasoning that connects perception to action, example reasoning that no robot is currently capable of. We include it because it illustrates the reasoning and problem solving abilities we would like to endow robots with and that we believe are worthy goals of Robotics and Artificial Intelligence, being within reach of both.

2. The need for intelligent robots

Where is the need for intelligent robots? Current (unintelligent) robots work fine so long as they are applied to simple tasks in almost predictable situations: parts of the correct type are presented in positions and orientations that hardly vary, and little dexterity is required for successful completion of the task. The huge commercial successes of robot automation have been of this sort: parts transfer (including palletizing and packaging), spot welding, and spray painting. Automation has been aimed largely at highly repetitive processes such as these in major industrial plants.

But to control the robot's environment sufficiently, it is typically necessary to erect elaborate fixtures. Often, the set-up costs associated with designing and installing fixtures and jigs dominate the cost of a robot application. Worse, elaborate fixturing is often not transferrable to a subsequent task, reducing the flexibility and adaptability that are supposedly the key advantages of robots. Sensing is one way to loosen up the environmental requirements; but the sensing systems of current industrial robots are mostly restricted to two-dimensional binary vision. Industrial applications requiring compliance, such as assembly, seam welding and surface finishing, have clearly revealed the inabilities of current robots. Research prototypes have explored the use of three-dimensional vision, force and proximity sensors, and geometric models of objects [Faugeras 1982, Clocksin et. al. 1982, Trevelyan, Kovesi, and Ong 1984, Nakagawa and Ninomiya 1984, Porter and Mundy 1982, 1984]. Other applications expose the limitations of robots even more clearly. The environment cannot be controlled for most military applications, including smart sentries, autonomous ordinance disposal, autonomous recovery of men and materiel, and, perhaps most difficult of all, autonomous navigation.

3. Roboticspart of ArtificialIntelligence

Artificial Intelligence (AI) is the field that aims to understand how computers can be made to exhibit intelligence. In any aspect of thinking, whether reasoning, perception, or action, the crucial questions are:

- *What knowledge is needed.* The knowledge needed for reasoning in relatively formalized and circumscribed domains such as symbolic mathematics and game playing is well known. Highly competent programs have been developed in such domains. It has proven remarkably difficult to get experts to precisely articulate their knowledge, and hence to develop programs with similar expertise, in medicine, evaluating prospective mining sites, or configuring computers (see [Michie 1979, Hayes-Roth, Waterman, and Lenat 1983, Winston 1983] for a discussion of expert systems, and accounts of the difficulty of teasing knowledge out of experts). Among the many severe inadequacies of the current crop of expert systems, is the fact that they usually have limited contact with the real world. Human experts perform the necessary perceptual preprocessing, telling MYCIN for example that the patient is "febrile 0.8". Moving from the restricted domain of the expert, to the unrestricted world of everyday experience, determining *what* knowledge is needed is a major

step toward modelling stereo vision, achieving biped walking and dynamic balance, and reasoning about mechanisms and space. What do you need to know in order to catch a ball?

- *Representing knowledge.* A key contribution of AI is the observation that knowledge should be represented *explicitly*, not heavily encoded, for example numerically, in ways that suppress structure and constraint. A given body of knowledge is used in many ways in thinking. Conventional data structures are tuned to a single set of processes for access and modification, and this renders them too inflexible for use in thinking. AI has developed a set of techniques such as semantic networks, frames, and production rules, that are symbolic, highly flexible encodings of knowledge, yet which can be efficiently processed.

Robotics needs to deal with the real world, and to do this it needs detailed geometric models. Perception systems need to produce geometric models; reasoning systems must base their deliberations on such models; and action systems need to interpret them. Computer-aided design (CAD) has been concerned with highly restricted uses of geometric information, typically display and numerically-controlled cutting. Representations incorporated into current CAD sytems are analogous to conventional data structures. In order to connect perception, through reasoning, to action, richer representations of geometry are needed. Steps toward such richer representations can be found in configuration space [Lozano-Pérez 1981, 1983a], generalized cones [Binford 1981], and visual shape representations [Horn 1982, Ikeuchi 1983, Brady 1984, Marr 1982].

As well as geometry, Robotics needs to represent forces, causation, and uncertainty. We know how much force to apply to an object in an assembly to mate parts without wedging or jamming [Whitney 1983]. We know that pushing too hard on a surface can damage it; but that not pushing hard enough can be ineffective for scribing, polishing, or fettling. In certain circumstances, we understand how an object will move if we push it [Mason 1983]. We know that the magnitude and direction of an applied force can be changed by pulleys, gears, levers, and cams.

We understand the way things such as zip fasteners, pencil sharpeners, and automobile engines work. The spring in a watch stores energy, which is released to a flywheel, causing it to rotate; this causes the hands of the watch to rotate by a smaller amount determined by the ratios of the gear linkages. Representing such knowledge is not simply a matter of developing the appropriate mathematical laws. Differential equations, for example, are a representation of knowledge that, while extremely useful, are still highly limited. Forbus [1983] points out that conventional mathematical representations do not encourage qualitative reasoning, instead, they invite numerical simulation. Though useful, this falls far short of the qualitative reasoning that people are good at. Artificial Intelligence research on qualitative reasoning and naive physics has made a promising start but has yet to make contact with the real world, so the representations and reasoning processes it suggests have barely been tested [Hobbs and Moore 1984, Forbus 1983, AI Journal 1984, DeKleer 1975, Winston, Binford, Katz, and Lowry 1984].

Robotics needs to represent uncertainty, so that reasoning can successfully overcome it. There are bounds on the accuracy of robot joints, feeders and sensors have errors; and though we talk about repetitive work, no two parts are ever exactly alike. As the tolerances on robot applications become tighter, the need to deal with uncertainty, and to exploit redundancy, becomes greater.

- *Using knowledge.* AI has also uncovered techniques for using knowledge effectively. One problem is that the knowledge needed in any particular case cannot be predicted in advance. Programs have to respond flexibly to a non-deterministic world. Among the techniques offered by AI are search, structure matching, constraint propogation, and dependency-directed reasoning. One approach to constraint propogation is being developed in models of perception by Terzopoulos [1983], Zucker, Hummel, and Rosenfeld [1977]. Another has been developed by Brooks [1981, 1982] building on earlier work in theorem proving. The application of search to Robotics has been developed by Goto et. al. [1980], Lozano-Pérez [1981], Gaston and Lozano-Pérez [1982], Grimson and Lozano-Pérez [1984], and Brooks [1983b]. Structure matching in Robotics has been developed by Winston, Binford, Katz, and Lowry [1984].

To be intelligent, Robotics programs need to be able to plan actions and reason about those plans. Surely AI has developed the required planning technology? Unfortunately, it seems that most, if not all, current proposals for planning and reasoning developed in AI require significant extension before they can begin to tackle the problems that typically arise in Robotics, some of which are discussed in Section 5. One reason for this is that reasoning and planning has been developed largely in conjunction with purely cognitive representations, and these have mostly been abstract and idealized. Proposals for knowledge representation have rarely been constrained by the need to support actions by a notoriously inexact manipulator, or to be produced by a perceptual system with no human preprocessing. ACRONYM [Brooks 1981, Brooks and Binford 1980] is an exeption to this criticism. Another reason is that to be useful for Robotics, a representation must be able to deal with the vagaries of the real world, its geometry, inexactness, and noise. All too often, AI planning and reasoning systems have only been exercised on a handful of toy examples.

In summary, Robotics challenges AI by forcing it to deal with real objects in the real world. Techniques and representations developed for purely cognitive problems, often in toy domains, do not necessarily extend to meet the challenge.

4. Action

In this section, we consider what is required for robots to act on their environment. This is the subject of kinematics and dynamics, control, robot arms, multi-fingered hands, and locomoting robots.

4.1. Kinematics, Dynamics, and Arm Design

The kinematics of robot arms is one of the better understood areas of Robotics [Paul 1981, Brady et. al. 1983]. The need for kinematic transformations arises because programmers prefer a different representation of the space of configurations of a robot than that which is most natural and efficient for control. Robots are powered by motors at the joints between links. Associated with a motor are quantities that define its position, velocity, acceleration, and torque. For a rotary motor, these are angular positions, angular velocities etc. It is most efficient to control robots in *joint space*. However, programmers prefer to think of positions using orthogonal, cylindrical, or spherical *Cartesian* coordinate frames, according to the task. Six degrees of freedom (DOF) are required to define the position and orientation of an object in space. Correspondingly, many robots have six joint motors to achieve these freedoms. Converting between the joint positions, velocities, and accelerations and the Cartesian (task) counterparts is the job of kinematics. The conversion is an identity transformation between the joint space of "Cartesian" arms (such as the IBM 7565) and orthogonal (x, y, z) Cartesian space. Cartesian arms suffer the disadvantage of being less able to reach around and into objects. Kinematic transformations are still needed to spherical or cylindrical Cartesian coordinates.

The kinematics of a mechanical device are *defined* mathematically. The requirement that the kinematics can be *efficiently computed* adds constraint, that ultimately affects mechanical design. In general, the transformation from joint coordinates to Cartesian coordinates is straightforward. Various efficient algorithms have been developed, including recent recursive schemes whose time complexity is linear in the number of joints. Hollerbach [1983] discusses such recursive methods for computing the kinematics for both the Lagrange and Newton-Euler formulations of the dynamics. The inverse kinematics computation, from Cartesian to joint coordinates, is often more complex. In general, it does not have a closed form solution [Pieper 1968]. Pieper [1968, see also Pieper and Roth 1969] showed that a "spherical" wrist with three intersecting axes of rotation leads to an exact analytic solution to the inverse kinematic equations. The spherical wrist allows a decomposition of the typical six degree of freedom inverse kinematics into two three degree of freedom computations, one to compute the position of the wrist, the other to compute the orientation of the hand. More recently, Paul [1981], Paul, Stevenson, and Renaud [1984], Featherstone [1983], and Hollerbach and Sahar [1983], have developed efficient techniques for computing the inverse kinematics for spherical wrists. Orin and Schrader [1984] have investigated algorithms for computing the Jacobian of the kinematic transformation that are suited to VLSI implementation.

If the number of robot joints is equal to six, there are singularities in the kinematics, that is, a small change in Cartesian configuration corresponds to a large change in joint configuration. The singularities of six degree-of-freedom industrial robot arms are well cataloged. Singularities can be avoided by increasing the number n of joints, but then there are infinitely many solutions to the inverse kinematics computation. One approach is to use a generalized inverse technique

using a positive definite b by r matrix to find the solution that minimizes some suitable quantity such as energy or time [Whitney 1968, Kahn and Roth 1971]. Another approach is to avoid singularities by switching between the redundant degrees of freedom [Paul (forthcoming)]. Finally, if the number of joints is less than six, there are "holes" in the workspace, regions that the robot cannot reach. Such robots, including the SCARA design, are nevertheless adequate for many specialized tasks such as pick-and-place operations. One important application of kinematics computations is in automatic planning of trajectories [Brady 1983c].

Most attention has centered on open kinematic chains such as robot arms. Much less work has been done on closed kinematic chains such as legged robots or multi-fingered hands. Hirose et. al. [1981] have designed a pantograph mechanism for a quadruped robot that significantly reduces potential energy loss in walking. Salisbury and Craig [1982] (see also Salisbury [1982]) have used a number of computational constraints, including mobility and optimization of finger placement, to design a three-fingered hand. The accuracy and dexterity of a robot varies with configuration, so attention needs to be paid to the layout of the workspace. Salisbury and Craig [1981] used the condition number of the Jacobian matrix (using the row norm) to evaluate configurations of the hand, that is to evaluate points in the workspace. Yoshikawa [1984] has introduced a *measure of manipulability* for a similar purpose. Roth [1984] reviews the application to Robotics of screw coordinates to link kinematics and dynamics.

The dynamic equations of a robot arm (see Hollerbach [1983]) consist of n coupled, second-order, differential equations in the positions, velocities, and accelerations of the joint variables. The equations are complex because they involve terms from two adjacent joints, corresponding to reaction and Coriolis torques. Conventional techniques have simplified dynamics by dropping or linearizing terms, or have proposed table look-up techniques. Recently, "recursive" recurrence formulations of the dynamic equations have been developed that:

1. Compute the kinematics from the shoulder to the hand in time proportional to n,

2. Compute the inverse dynamics from the force and torque exerted on the hand by the world from the hand to the shoulder, again in time proportional to n.

The importance of this result is threefold:

- First, it suggests that a more accurate inverse plant model can be developed, leading to faster, more accurate arms. Friction is a major source of the discrepancy between model and real world. Direct drive technology [Asada and Kanade 1981, Asada 1982, Asada and Youcef-Toumi 1984] reduces the mismatch. In a direct drive arm, a motor is directly connected to a joint with no intervening transmission elements, such as gears, chains, or ball screws. The advantages are that friction and backlash are low, so the direct drive joint is backdrivable. This means that it can be controlled using torque instead of position. Torque control is important for achieving compliance, and for feedforward dynamics compensation.

- Second, the recurrence structure of the equations lends itself to implementation using a pipelined microprocessor architecture, cutting down substantially on the number of wires that are threaded through the innards of a modern robot.

- Third, Hollerbach and Sahar [1983] have shown that their refinement of Featherstone's technique for computing the inverse kinematics makes available many of the terms needed for the recursive Newton-Euler dynamics.

Renaud [1984] has developed a novel iterative Lagrangian scheme that requires about 350 additions and 350 multiplies for a six revolute joint robot arm. The method has been applied to manipulators having a general tree structure of revolute and prismatic joints. Huston and Kelly [1982] and Kane and Levinson [ ] have recently adapted Kane's formulation of dynamics to robot structures.

4.2. Control

Much of control theory has developed for slowly changing, nearly rig ' ,stems. The challenges of robot control are several:

- *Complex dynamics.* The dynamics of open-link kinematic chain robots consist of n coupled second-order partial differential equations, where n is the number of links. They become even more complex for a closed multi-manipulator system such as a multi-fingered robot hand or locomoting robot.

- *Articulated structure.* The links of a robot arm are cascaded and the dynamics and inertias depend on the configuration.

- *Discontinuous change.* The parameters that are to be controlled change discontinuously when, as often happens, the robot picks an object up.

- *Range of motions.* To a first approximation one can identify several different kinds of robot motion: free space or gross motions, between places where work is to be done; approach motions (guarded moves) to a surface; and compliant moves along a constraint surface. Each of these different kinds of motion poses different control problems.

The majority of industrial robot controllers are open-loop. However, many control designs have been investigated in Robotics; representative samples are to be found in [Brady et. al. 1983, Paul 1981, Brady and Paul 1984]. They include optimal controllers [Kahn and Roth 1971, Dubowsky 1983]; model reference control [Dubowsky and des Forges 1979]; sliding mode control [Young 1978 (in Brady et. al. 1983)]; non-linear control [Freund 1981, 1984]; hierarchical control [Salisbury and Craig 1981]; distributed control [Klein and Wahawisan, 1982]; hybrid force-position control [Raibert and Craig 1981 (reproduced in Brady et. al. 1983), Klein Olson, and Pugh 1983]; and integrated system control [Albus 1983]. Cannon and Schmitz [1984] have investigated the precise control of flexible manipulators.

4.3. End effectors

Industrial uses of robots typically involve a multi-purpose robot arm and an end effector that is specialized to a particular application. End effectors normally

have a small degree of freedom (Crossley and Umholtz 1980), while grippers such as the cup, spatula, fork, and cook (Hall et al. 1980) have no degrees of freedom. The algorithms for using such grippers are correspondingly simple (Paul 1972, see also Taylor, Summers, and Meyer 1982 centering grasp), but this is one of the more advanced examples. Many end effectors have no built in sensors. Those that do typically incorporate devices that give a single bit of information. The most common are contact switches and infra red beams to determine when the end effector is spanning some object. The IBM 7565 is one of the few commercially available robot arms that incorporates force sensing and provides linguistic support for it.

Many tasks, particularly those related to assembly, require a variety of capabilities, such as parts handling, insertion (Whitney 1983), screwing, and variable fixturing. One approach is to use one arm but multiple single DOF grippers, or multiple arms each with a single DOF, or some combination. One problem with using multiple single DOF grippers is that a large percentage of the work cycle is spent changing grippers. This has inspired research on the mechanical designs that support fast gripper change. Another problem is that the approach assumes that a process can be divided into a discrete set of single DOF operations.

Multiple arms raise the problem of coordinating their motion while avoiding collision and without cluttering the workspace. The coordination of two arms was illustrated at the Stanford Artificial Intelligence Laboratory in 1972 when two arms combined to install a hinge. One of the arms performed the installation, the other acted as a programmable fixture, presenting the hinges and other parts to the work arm. Freund [1984] has presented a control scheme for preventing collisions between a pair cooperating robots. Whenever there is a possibility of a collision, one of the arms is assigned master status and the other one has to modify its trajectory to avoid the master.

In contrast with such end effectors, a human hand has a remarkable range of functions. The fingers can be considered to be sensor intensive 3 or 4 DOF robot arms. The motions of the individual fingers are limited to curl and flex motions in a plane that is determined by the abduction/adduction of the finger about the joint with the palm. The motions of the fingers are coordinated by the palm, which can assume a broad range of configurations. The dexterity of the human hand has inspired several researchers to build multi-function robot hands.

Okada [1979] described a hand consisting of three fingers evenly spaced about a planar palm. The work-space of the individual fingers was an ellipsoid. The three workspaces intersected in a point. Okada programmed the hand to perform several tasks such as tighten bolts. Hanafusa and Asada [1977, 1979] developed a hand consisting of three evenly-spaced, spring-loaded fingers. The real and apparent spring constants of the fingers were under program control. Stable grasps were defined as the minima of a potential function. The definition of stability in two dimensions was demonstrated by programming the hand to pick up an arbitrary shaped lamina viewed by a TV camera.

Salisbury [1982, see also Salisbury and Craig 1982] investigated kinematic and force constraints on the design of a tendon-driven three-fingered hand (see Figure 1).

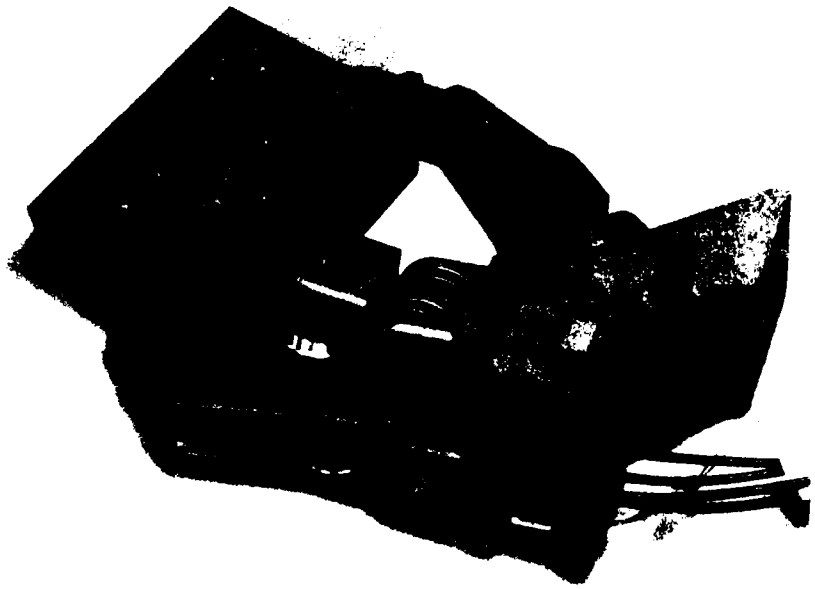


Figure 1. The three-fingered Robot hand developed by Salisbury and Craig [1981]. Each finger has three degrees of freedom, and is pulled by four tendons. The hierarchical controller includes three finger controllers, each of which consists of four PID controllers, one per tendon. Reproduced from [Salisbury 1982]

The goal was to design a hand that could impress an arbitrary (vector) force in an arbitrary position of the hand's workspace. Four of the parameters defining the placement of the thumb were determined by solving a series of one-dimensional nonlinear programming problems. A hierarchical controller was designed: PID controllers for each tendon; four such for each finger; and three such for the hand. To date, position and force controllers have been implemented for the individual fingers. A force sensing palm has recently been developed [Salisbury 1984]. It can determine certain attributes of contact geometries. The force sensing fingertips being developed for the hand will permit accurate sensing of contact locations and surface orientations. This information is likely to be useful in object recognition strategies and in improving the sensing and control of contact forces.

The Center for Biomedical Design at the University of Utah and the MIT Artificial Intelligence Laboratory are developing a tendon-operated, multiple DOF robot hand with multi-channel touch sensing. The hand that is currently being built consists of three 4 DOF fingers, a 4 DOF thumb, and a 3 DOF wrist (total 19 DOF). Three fingers suffice for static stable grasp. The Utah-MIT design incorporated a fourth finger to minimize reliance on friction and to increase flexibility in grasping tasks. The hand features novel tendon material and tendon routing geometry (Figure 2).

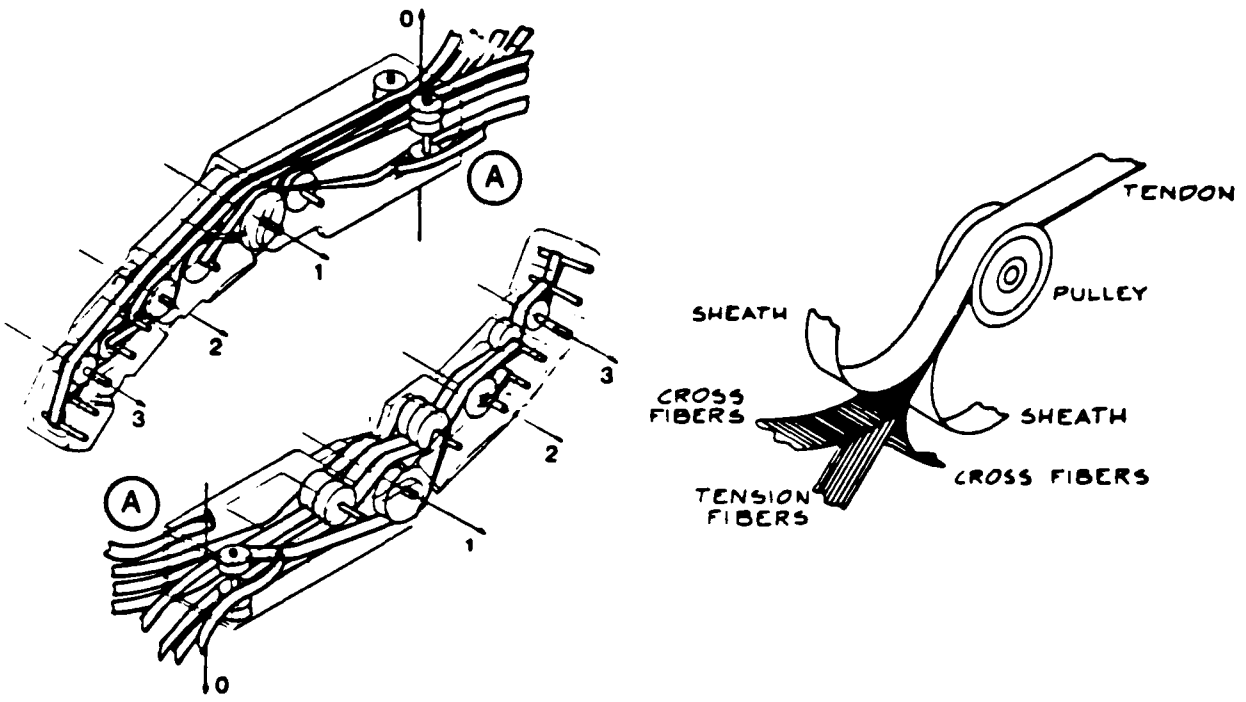


Figure 2. The prototype Utah/MIT dextrous hand developed by Steven Jacobsen, John Wood, and John Hollerbach. The four fingers each have four degrees of freedom. a. The geometry of tendon routing. b. The material composition of tendons. (Reproduced from [Jacobsen *et. al* 1984, Figures 2b and 7])

4.4. Mobile robots

Certain tasks are difficult or impossible to perform in the workspace of a static robot arm [Giralt 1984]. In large scale assembly industries, such as shipyards or automobile assembly lines, it is common to find parts being transferred along gantries that consist of one or more degrees of linear freedom. Correspondingly, there have been several instances of robot arms being mounted on rails to extend their workspace. The rail physically constrains the motion of the robot. More generally, the robot can be programmed to follow a path by locally sensing it. Magnetic strips, and black strips sensed by infra-red sensing linear arrays, have been used, for example in the Fiat plant in Turin, Italy.

More ambitious projects have used (non-local) vision and range data for autonomous navigation. Space and military applications require considerable autonomy for navigation, planning, and perception. Mobile robots are complex systems that incorporate perceptual, navigation, and planning subsystems. Shakey [Nilsson 1969] was one of the earliest mobile robots, and certainly one of the most ambitious system integration efforts of its time. Later work on mobile robots includes [Dobrotin and Lewis 1975, Giralt, Sobek, and Chatila 1977, Giralt, Chatila, and Vaisset 1984, Lewis and Johnson 1977, Lewis and Bejczy 1973, Harmon 1983a, 1983b, Everett 1982, Moravec 1981, 1983, 1984].

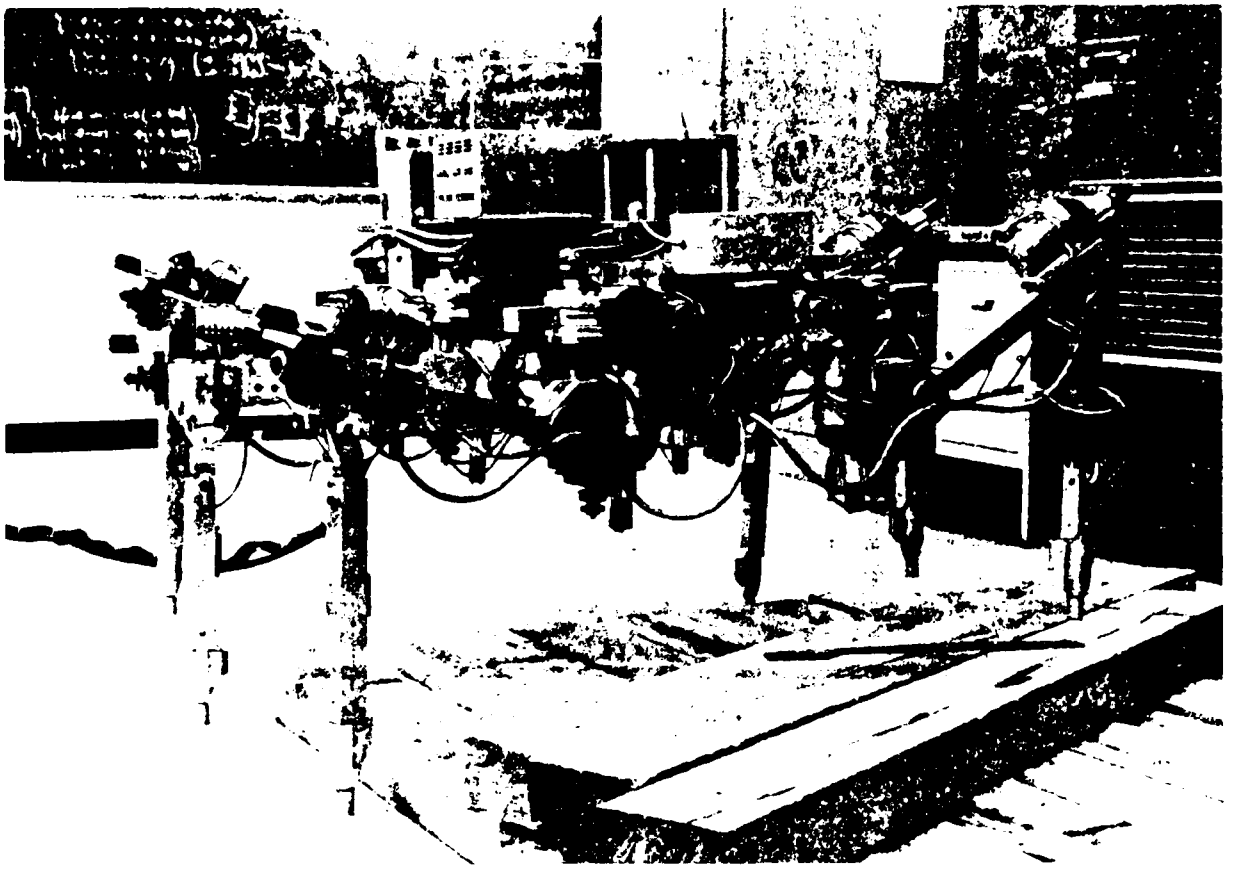


Figure 3. The six legged robot developed by McGhee, Orin, Klein, and their colleagues at the Ohio State University. The robot uses either an alternating tripod of support or wave gait. (Reproduced from Ozguner, Tsai, and McGhee 1984, Figure 1)

All the robots referred to previously in this section are wheeled. This restricts their movement to (nearly) even terrain. Legged vehicles can potentially escape that limitation. The price to be paid for this advantage is the extra complexity of maintaining balance and controlling motion. Following the photographic studies of Muybridge and Moayer in the late 19th century, a theory of locomotion developed around the concept of *gait*, the pattern of foot placements and foot support duty cycles. The OSU hexapod, for example, has been programmed to use either an alternating tripod of support or "wave" gait, in which a left right pair of legs is lifted and advanced (Figure 3) (Ozguner, Tsai, and McGhee 1984). Hirose's [1984] quadruped robot (Figure 4) employs a "crab" gait, in which the direction of motion is at an angle to the direction the quadruped is facing.

Two generic problems in legged locomotion are moving over uneven terrain and achieving dynamic balance without a static pyramid of support.

The simplest walking machines employ a predetermined, fixed gait. Uneven terrain requires dynamic determination of foot placement, implying variable gait. Hirose

Figure 4. The quadruped robot built by Hirose, Umetani, and colleagues at Tokyo Institute of Technology. The robot can walk upstairs, and can move forward in a crab gait. (Reproduced from [Hirose *et. al.* 1984, photo 1])

[1984] and Ozguner, Tsai, and McGhee [1984] analyze the constraint of balance, and use sensors in their choice of foot placement.

Miura and Shimoyama [1984] and Raibert et. al. [1982, 1984] discuss the dynamic requirements of balance. Miura and Shimoyama reports a series of biped walking machines. The first of these, BIPER-3 (Figure 5) has stilt-like legs, with no ankle joint, and resembles a novice nordic skier in its gait. BIPER-3 falls if both feet keep contact with the surface; so it must continue to step if it is to maintain its balance. An ambitious development, BIPER-4, shown in Figure 6, has knee and ankle joints. Stable walking of BIPER-4 has recently been demonstrated. Raibert considers balance for a three-dimensional hopping machine (Figure 7). He suggests that balance can be achieved by a planar (two-dimensional) controller plus extra-planar compensation. His work suggests that gait may not be as central to the theory of locomotion as has been supposed. Instead, it may be a side-effect of achieving balance with coupled oscillatory systems. Raibert [1984] has organized a collection of papers on legged robots that is representative of the state of the art.

5. Reasoning about objects and space

5.1. Task-level robot programming languages

Earlier, we listed some of the software features of modern, commercially available robot systems: they perform trajectory calculation and kinematic translation, interpret sense data, execute adaptive control through conditional execution and real time monitors, interface to databases of geometric models, and support program development. Despite these features, robot programming is tedious, mostly because

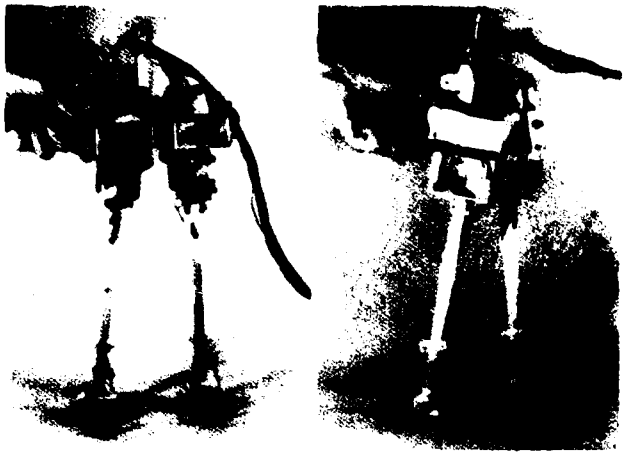


Figure 5. The BIPER-3 walking robot built by Miura and Shimoyama at Tokyo University. See text for details. (Reproduced from [Miura and Shimoyama 1984, Figure 1])

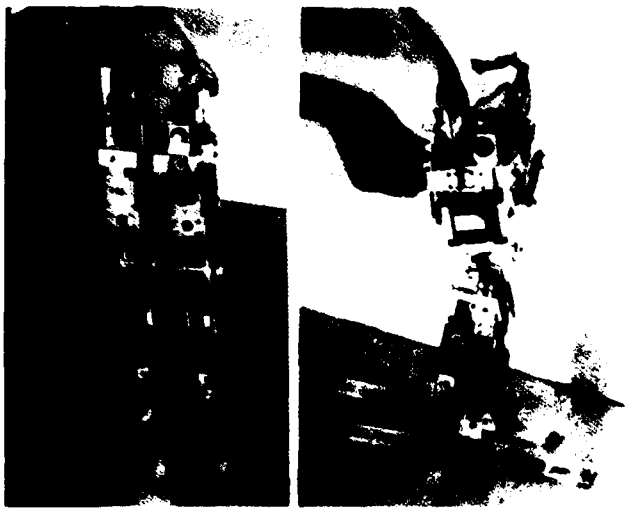


Figure 6. The BIPER-4 walking robot under development at Tokyo University. BIPER-4 has a hip, knee, and ankle joints on each leg. (Reproduced from [Miura and Shimoyama 1984, Figure 14])

in currently available programming languages the position and orientation of objects, and subobjects of objects, have to be specified exactly in painful detail. "Procedures" in current robot programming languages can rarely even be parameterized, due to physical assumptions made in the procedure design. Lozano-Pérez [1983b, 1983c] calls such programming languages *robot level.*

Lozano-Pérez [1983, page 839] suggests that "existing and proposed robot programming systems fall into three broad categories: *guiding* systems in which the user leads a robot through the motions to be performed, *robot-level* programming

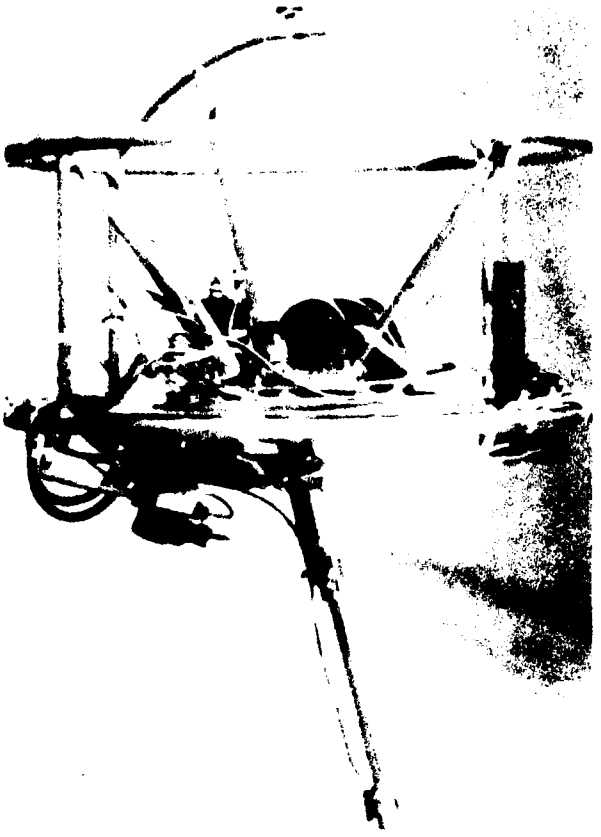


Figure 7. The hopping machine developed by Raibert and his colleagues at Carnegie Mellon University. (Reproduced from [Raibert *et. al.* 1984, Figure 16])

systems in which the user writes a computer program specifying motion and sensing, and *task-level* programming systems in which the user specifies operations by their desired effect on objects." Languages such as VAL II [Shimano, Geschke, and Spaulding 1984] and AML [Taylor, Summers, and Meyer 1982] are considered robot-level.

One of the earliest task-level robot programming language designs was AUTOPASS [Liebermann and Wesley 1977]. The (unfinished) implementation focussed upon planning collision-free paths among polyhedral objects. The emphasis of RAPT [Ambler and Popplestone 1975, Popplestone, Ambler, and Bellos 1980] has been on the specification of geometric goals and relational descriptions of objects. The implementation of RAPT is based upon equation solving and constraint propagation. Other approximations to task-level languages include PADL [Requicha 1980], IBMsolid [Wesley et. al. 1980], and LAMA [Lozano-Pérez 1976]. Lozano-Pérez [1983c] discusses spatial reasoning and presents an example of the use of RAPT.

In the next section we discuss Brooks' work on uncertainty, several approaches to reasoning about space and avoiding obstacles, and synthesizing compliant programs.

5.2. Dealing with uncertainty

Consider the problem illustrated in Figure 8. A robot has been programmed to put a screw in a hole. Will the program succeed? Each of the joint measurements of the robot is subject to small errors, which produce errors in the position and orientation of the finger tips according to the Jacobian of the kinematics function. The position and orientation of the screwdriver in the fingers is subject to slight error, as is the screw, box, and the lid on the box. These errors, we will call them the *base errors*, are independent of the particular task to be performed. They add up. Taylor [1976] assumed particular numerical bounds for the values of the base errors, and used linear programming to bound the error in the placement of the screw relative to the hole.

Brooks [1982] worked with explicit symbolic (trigonometric) expressions that define the error in the placement of the screw relative to the hole. He applied the expression bounding program developed for the ACRONYM project [Brooks 1981] to the base error bounds used by Taylor to deduce bounds for the errors in the placement of the screw relative to the hole. The bounds he obtained were not as tight as those obtained by Taylor, but were nearly so.

Brooks' approach had a substantial advantage over Taylor, however, and it is paradigmatic of the AI approach. The expression bounding program can be applied with equal facility to the symbolic expression for the error and the desired size of the screw hole (the specifications of the insertion task). The result is a bound on the only free variable of the problem, the length of the screwdriver. The lesson is that it is possible to apply AI techniques to reason in the face of uncertainty. In further work, Brooks [1982] has shown how sensing might be modeled using uncertainties to automatically determine when to splice a sensing step into a plan to cause it to succeed.

5.3. Reasoning about space and avoiding objects

Robot-level programming languages require the programmer to state, for example, that the robot is to move the block B, whose configuration (position and orientation) $\mathbf{R}_S$ is to be moved to the configuration $\mathbf{R}_G$. To ensure the robot does not crash into obstacles, the usual practice in robot level languages is to specify a sufficient number of via points (Figure 9) (see [Brady 1983c]). In a task oriented language, one merely says something like "put B in the vise". It follows that a crucial component of implementing a task oriented programming language is automatically determining safe paths between configurations in the presence of obstacles. This turns out to be an extremely hard problem.

Lozano-Pérez [1983a] introduced a representation called *C-space* that consists of the safe configurations of a moving object. For an single object moving with 6 degrees of freedom (eg., 3 translational and 3 rotational degrees of freedom), the dimensionality of the C-space is six. If there are m such objects, each of which can move, the dimensionality of C-space is $6m$. For example, for the coordinated motion of two 3D objects, C-space is twelve dimensional. In practice one can deal with "slices", projections onto lower dimensional subspaces.

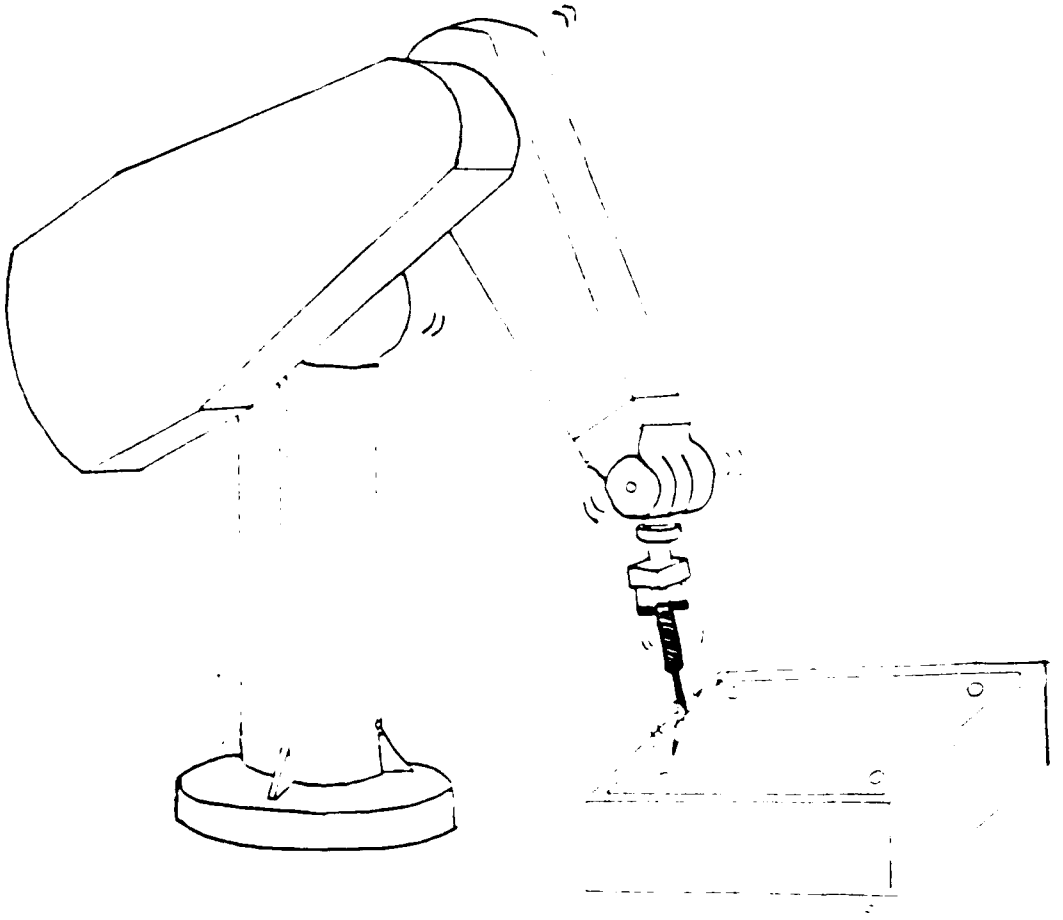


Figure 8. Will the screw make it into the hole?

Donald [1984] notes that there are two components of spatial planning systems. First, it is necessary to represent the problem, in particular the obstacles. Second, it is necessary to devise an algorithm that can search for paths over the representation. Most work on spatial reasoning has used representations that approximate the exact polyhedral obstacles. Such representations may (1) restrict the degrees of freedom in a problem, (2) bound objects in real-space by simple objects such as spheres, or prisms with parallel axes, while considering some subset of the available degrees of freedom, (3) quantize configuration space at certain orientations, or (4) approximate swept volumes for objects over a range of orientations. Systems that use such representations may not be capable of finding solutions in some cases, even if they use a complete search procedure. An approximation of the obstacle environment, robot model, or C-space obstacles can result in a transformed find-path problem which has no solution.

Lozano-Pérez [1981, 1983] implemented an approximate algorithm for Cartesian manipulators (for which free space and C-space are the same) that tesselated free space into rectangloids, subdividing it as far as necessary to solve a given problem. The search algorithm is complete for translations, and illustrates the feasibility of

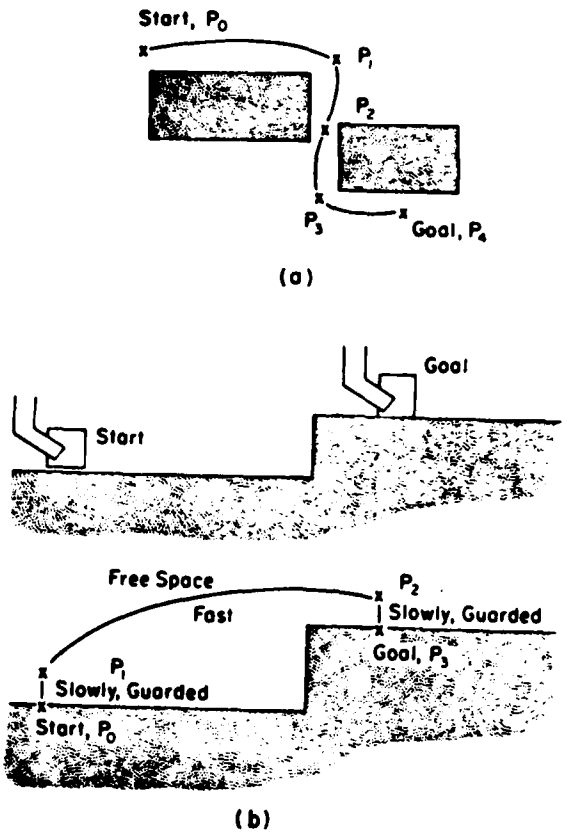


Figure 9. Point P_1, P_2, and P_3 are specified as via points to coax the robot through the narrow gap separating the two obstacles shown. (Reproduced from [Brady 1983c, Figure 2a])

the C-space approach. It works by alternately keeping the heading of an object fixed and rotating in place to alter the heading. Recently, Brooks and Lozano-Pérez [1983] reported an algorithm capable of moving a reorientable polygon through two-dimensional space littered with polygons. This algorithm can find any path of interest for the two-dimensional problem. Figure 10 shows an example path found by Brooks and Lozano-Pérez's program. Their attempts to extend the method to three dimensions "were frustrated by the increased complexity for three dimensional rotations relative to that of rotations in two dimensions" [Brooks 1983b, p7].

Brooks [1983a] suggested that free space be represented by overlapping generalized cones that correspond to *freeways* or *channels*. Figure 11 shows some of the generalized cones generated by two obstacles and the boundary of the workspace. The key point about the representation was that the left and right radius functions defining a freeway could be inverted easily. Given a freeway, and the radius function of a moving convex object, he was able to determine the legal range of orientations that ensure no collisions as the object is swept down the freeway. Brooks' algorithm is highly efficient, and works well in relatively uncluttered space, but it occasionally fails to find a safe path when it is necessary to maneuver in tight spaces. Recently, Donald [1983] has proposed a novel channel based technique.

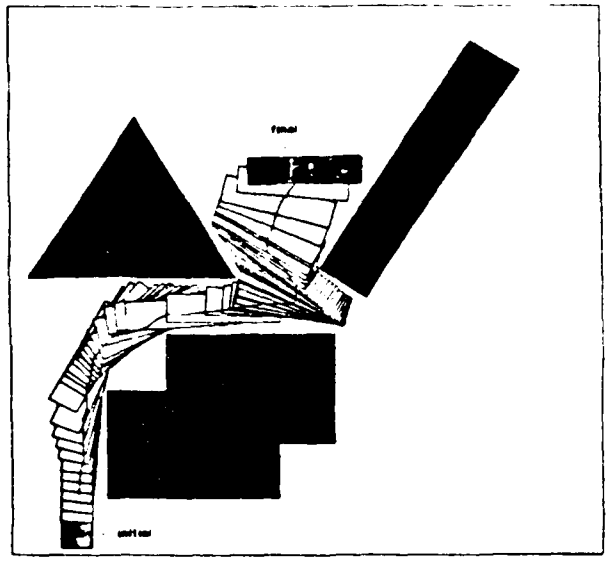


Figure 10. A path found by the Brooks and Lozano-Pérez program. (Reproduced from [Brooks and Lozano-Pérez 1983, Figure 11a])

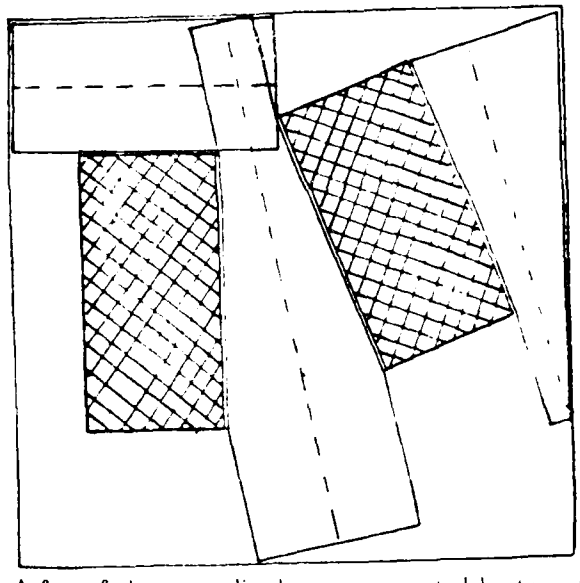


Figure 11. A few of the generalized cones generated by two obstacles and the boundary of the workspace. (Reproduced from [Brooks 1983a, Figure 1])

Finally, Brooks [1983b] has developed an algorithm that combines the C-space and freeway approaches to find paths for pick and place and insertion tasks for a PUMA. Pick and place tasks are defined as four degree of freedom tasks in which the only reorientations permitted are about the vertical, and in which the found path is composed of horizontal and vertical straight lines. Figure 12 shows an example path found by Brooks' algorithm.

Brooks freezes joint 4 of the PUMA. The algorithm subdivides free space to find (i) freeways for the hand and payload assembly, (ii) freeways for the upper arm subassembly (joints 1 and 2 of the PUMA); (iii) searches for the payload and

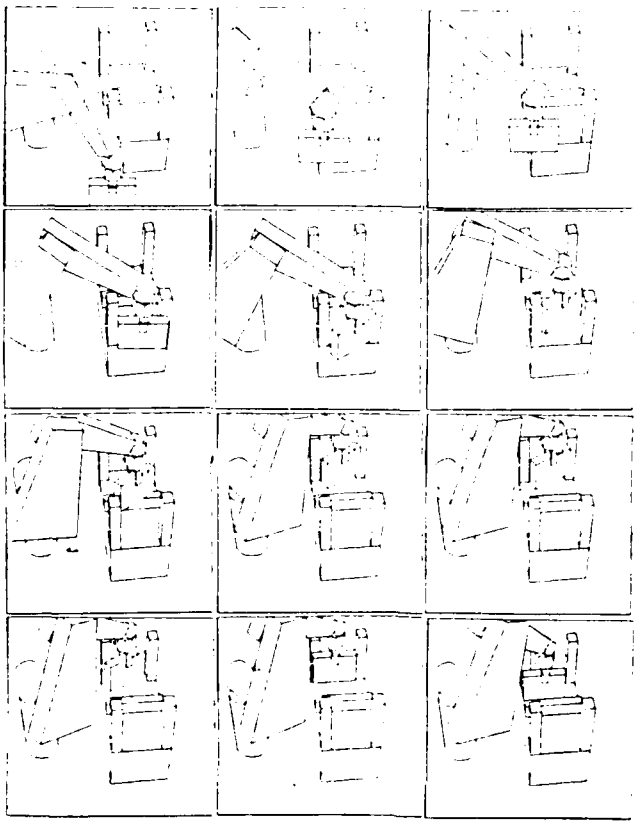


Figure 12. An example of path finding for a PUMA by Brooks' [1983b] algorithm.

upper arm freeways concurrently under the projection of constraints determined by the forearm. The subdivision of free space in this way is the most notable feature of Brooks' appoach. It stands in elegant relation to the algorithms for computing inverse kinematics referred to earlier. It is assumed that the payload is convex, and that the obstacles are convex stalagmites and stalactites. It is further assumed that stalactites are in the workspace of the upper arm of the PUMA, not of the payload.

By restricting attention to a limited class of tasks, Brooks has designed an efficient algorithm will not work in all cases. The advantage is that he does not have to contend with worst case situations that lead to horrendous polynomial complexity estimates. For example, Schwartz and Sharir [1983] suggest a method whose complexity for r DOF is n^{2^r}. For example, for two moving objects, the complexity is n^{4096}. Their algorithm is not implemented. Donald [1984] presents the first implemented, representation-complete, search-complete algorithm (at a given resolution) for the classical Movers' problem for Cartesian manipulators.

There has been a great deal of theoretical interest in the findpath problem by researchers in computational complexity and computational geometry. Schwartz and Sharir [1983], and Hopcroft, Schwartz, and Sharir [1983] are representative.

5.4. Synthesizing compliant programs

Compliance is the opposite of stiffness. Any device, such as an automobile shock absorber, that responds flexibly to external forces is called compliant. Increasingly, compliance refers to operations that require simultaneous force and position control. Mason 1983]. Force analysis of the peg in hole problem and the subsequent development of the Remote Center Compliance (RCC) [Whitney, 1983] is an important example of the use of force trajectories to achieve compliant assembly. Another example is scribing a straight line on an undulating surface. In that case, it is necessary to control position in the tangent plane of the surface, and maintain contact with the surface by applying an appropriate scribing force normal to the surface. Other examples of compliance include cutting, screw insertion, and bayonet-style fixtures, such as camera mountings. Clocksin *et. al.* 1982] describe a seam welding robot that uses the difference-of-Gaussians edge operator proposed by Marr and Hildreth [1980] to determine the welding trajectory. Ohwovoriole and Roth [1981] show how the motions possible at an assembly step can be partitioned into 3 classes: those that tend to separate the bodies to be mated; those that tend to make one body penetrate the other; and those that move the body and maintain the original constraints. Theoretically, this provides a basis for choosing the next step in an assembly sequence.

Trevelyan, Kovesi, and Ong [1984] describe a sheep shearing robot. Figure 13 shows the geometry of the robot and the "workpiece". Trevelyan, Kovesi, and Ong [1984] note that "over two hundred sheep have been shorn by the machine (though not completely) yet only a few cuts have occurred. This extremely low injury rate[1] results from the use of sensors mounted on the shearing cutter which allow the computer controlling the robot to keep the cutter moving just above the sheep skin". Trajectories are planned from a geometric model of a sheep using Bernstein-Bézier parametric curves. The trajectory is modified to comply with sense data. Two capacitance sensors are mounted under the cutter just behind the comb. These sensors can detect the distance between the cutter and the sheep's skin to a range of approximately 30mm. Compliance is needed to take account of inaccuracies in the geometric model of the sheep and the change in shape of the sheep as it breathes.

Robots evolved for positional accuracy, and are designed to be mechanically stiff. High tolerance assembly tasks typically involve clearances of the order of a thousandth of an inch. In view of inaccurate modeling of the world and limitations on joint accuracy, low stiffnesses are required to effect assemblies. Devices such as the Remote Center Compliance (RCC) [Whitney 1983] and the Hi-T hand [Goto 1980] exploit the inherent mechanical compliance of springs to accomplish tasks. Such *passive compliant* devices are fast, but the specific application is built into the mechanical design. The challenge is that different tasks impose different stiffness requirements. In *active compliance*, a computer program modifies the trajectory of the arm on the basis of sensed forces (and other modalities) [Paul and Shimano

[1] Trevelyan informs the author that a human sheep shearer typically cuts a sheep every 30 minutes, and that serious cuts occur regularly.

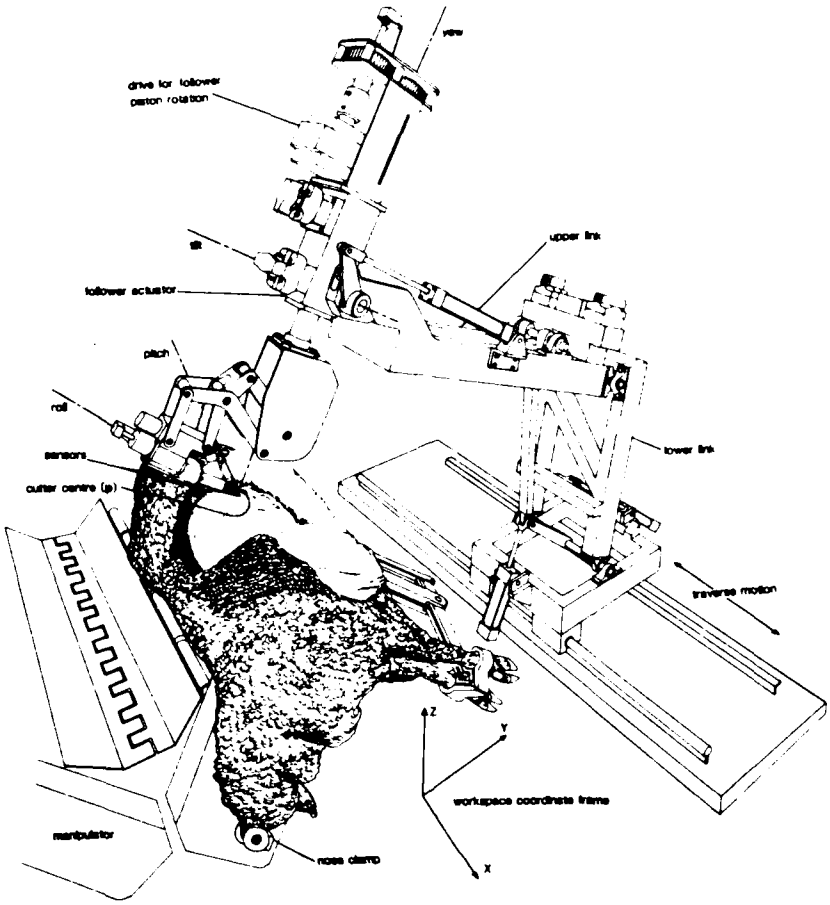


Figure 13. The sheep shearing robot developed by Trevelyan and his colleagues at the University of Western Australia. Sheep naturally lies quite still while it is being sheared; indeed it often falls asleep. (Reproduced from [Trevelyan, Kovesi, and Ong 1984, figure 1])

1976]. Active compliance is a general purpose technique; but is typically slow compared to passive compliance.

Mason [1981] suggested that the (fixed number of) available degrees of freedom of a task could be divided into two subsets, spanning orthogonal subspaces. The subspaces correspond one-one with the *natural constraints* determined by the physics of the task, and the *artificial constraints* determined by the particular task. See [Mason 1983] for details and examples. For example, in screwdriving, a screwdriver cannot penetrate the screw, giving a natural constraint; successful screwdriving requires that the screwdriver blade be kept in the screw slot, an artificial constraint. Raibert and Craig [1983] refined and implemented Mason's model as a hybrid force-position controller. Raibert and Craig's work embodies the extremes of stiffness control in that the programmer chooses which axes should be controlled with infinite stiffness (using position control with an integral term) and which should be controlled with zero stiffness (to which a bias force is added). Salisbury [1980] suggests an intermediate ground that he calls "active stiffness control".

Programmers find it relatively easy to specify positions in free space but find it hard to specify the force based trajectories needed for compliance. This has motivated the investigation of automatic generation of compliant fine motion programs [Dufay and Latombe 1984, Lozano-Perez, Mason and Taylor 1984]. In Dufay and Latombe's approach, the geometry of the task is described by a semantic network, the initial and goal configurations of parts are described by symbolic expressions, and the knowledge of the program is expressed as production rules. Productions encode the "lore" of assembly: how to overcome problems such as moving off the chamfer during an insertion task. Dufay and Latombe's program inductively generates assembly programs from successful execution sequences. The method requires that the relationships between surfaces of parts in contact be known fairly precisely. In general this is difficult to achieve because of errors in sensors.

Lozano-Pérez, Mason, and Taylor [1983] have proposed a scheme for automatically synthesizing compliant motions from geometric descriptions of a task. The approach combines Mason's ideas about compliance, Lozano-Pérez's C-space, and Taylor's [1976] proposal for programming robots by fleshing out skeletons forming a library of operations. The approach, currently being implemented, deals head-on with errors in assumed position and heading. Lozano-Pérez, Mason, and Taylor use a generalized damper model to determine all the possible configurations that can result from a motion. It is necessary to avoid being jammed in the friction cone of any of the surfaces *en route* to the goal surface. This sets up a constraint for each surface. Intersecting the constraints leaves a range of possible (sequences of) compliant moves that are guaranteed to achieve the goal, notwithstanding errors.

Programming by fleshing out skeletons is reminiscent of the *programmer's apprentice* [Rich and Waters 1981]. The similarities are that the computer adopts the role of junior partner or critic, programming is based on *cliches*, and design decisions and logical dependencies are explicitly represented so that the effects of modifications to a program can be automatically propogated through the program. The difference is that a *robot* programmer's apprentice works with rich geometric models. Lozano-Pérez has suggested that guiding can be extended to teach a robot plans that involve sensing, a large number of similar movements (for example unloading a palette), and asynchronous control of multiple manipulators. The requirement that a system deal with rich geometric models also distinguishes the robot programmer's apprentice from earlier work in AI planning [Sacerdoti 1975].

6. Perception

6.1. Introduction

The perceptual abilities of commercially available robots are severely limited, especially when compared with laboratory systems. It is convenient to distinguish contact and non-contact sensing. Contact, or local, sensing includes tactile,

proximity, and force sensing. Non-contact sensing includes passive sensing in both visual and non visual spectral bands, and active sensing using infra red, sonar, ultrasound, and millimeter radar.

Robot perception is only a special case of *computer* perception in the sense that there are occasional opportunities for engineering solutions to what are, in general, difficult problems. Examples include: arranging the lighting, controlling positional uncertainty, finessing some of the issues in depth computation, and limiting the visual context of an object. Appropriate lighting can avoid shadows, light striping and laser range finding can produce partial depth maps, and techniques such as photometric stereo [Woodham 1981] can exploit control over lighting. On the other hand, edge finding is no less difficult in industrial images, texture is just as hard, and the bin of parts is a tough nut for stereo. Motion tracking on a dirty conveyor belt is as hard as any other tracking problem. Representing the shape of complex geometric parts is as difficult as any representational problem in computer vision [Faugeras et. al. 1983, 1984, Brady and Asada 1984, Bolles *et. al.* 1984]. Existing commercial robot vision systems carry out simple inspection and parts acquisition. There are, however, many inspection, acquisition, and handling tasks, routinely performed by humans, that exceed the abilities of current computer vision and tactile sensing research.

The quality of sensors is increasing rapidly, especially as designs incorporate VLSI. The interpretation of sensory data, especially vision, has significantly improved over the past decade. Sensory data interpretation is computer intensive, requiring billions of cycles. However, much of the computer intensive early processing naturally calls for local parallel processing, and is well suited to implementation on special purpose VLSI hardware [Brady 1983a, Raibert and Tanner 1982].

6.2. Contact sensing

Contact sensing is preferred when a robot is about to be, or is, in contact with some object or surface. In such cases, objects are often occluded, even when a non-contact sensor is mounted on a hand. An exception to this is seam welding [Clocksin et. al. 1982]. The main motivation for force sensing is not, however, to overcome occlusion, but to achieve compliant assembly. Force sensors have improved considerably over the past two or three years. Typical sensitivities range from a half ounce to ten pounds. Most work on force trajectories has been application specific (eg peg-in-hole insertion). Current research, aimed at developing general techniques for interpreting force data and synthesizing compliant programs, were discussed in the previous section. Kanade and Sommer [1984] and Okada [1982] report proximity sensors.

Touch sensing is currently the subject of intensive research. Manufacturing engineers consider tactile sensing to be of vital importance in automating assembly [Harmon 1982, 1984]. Unfortunately, current tactile sensors leave much to be desired. They are prone to wear and tear, have poor hysteresis, and low dynamic range. Industrially available tactile sensors typically have a spatial resolution of only about 8 points per inch. Tactile sensors are as poor as TV cameras were in

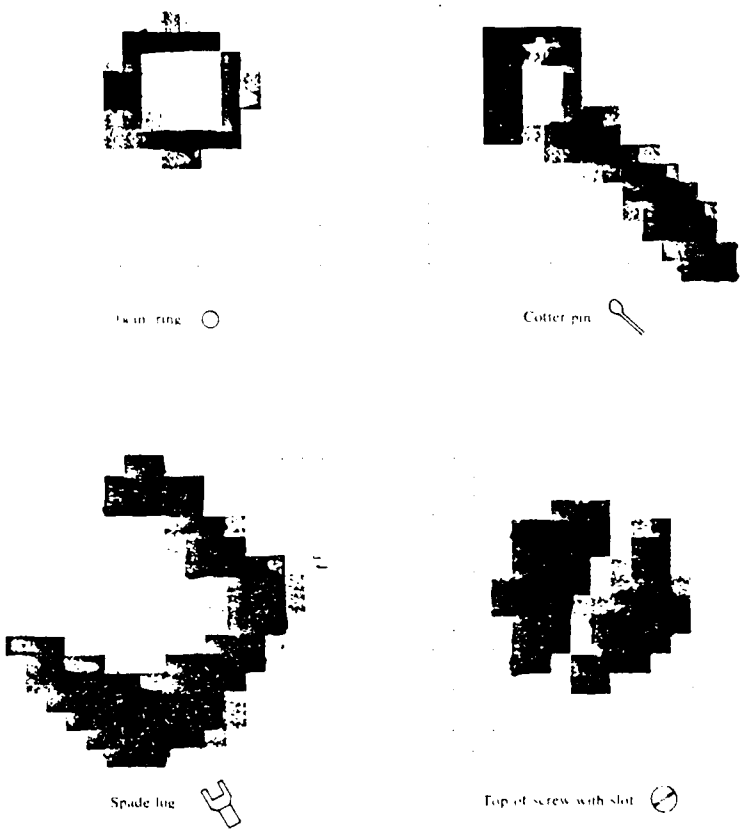


Figure 14. Sample tactile images from the Hillis sensor. (Reproduced from [Hillis 1982, Figure 6])

the 1960s, the analogy being that they are seriously hampering the development of tactile interpretation algorithms.

Several laboratory demonstrations point the way to future sensors. Hillis [1982] devised a tactile sensor consisting of an anisotropic silicon conducting material whose lines of conduction were orthogonal to the wires of a printed circuit board and which were separated by a thin spacer. Figure 14 shows some example touch images generated by Hillis' tactile sensor for four small fasteners. The sensor had a spatial resolution of 256 points per square *centimeter*. Raibert and Tanner [1982] developed a VLSI tactile sensor that incorporated edge detection processing on the chip (Figure 15). This (potentially) significantly reduces the bandwidth of communication between the sensor and the host computer. Recently, Hackwood and Beni [1983] have developed a tactile sensor using magneto-restrictive materials that appears to be able to compute shear forces.

Little progress has been made in the development of tactile object recognition algorithms. Hillis built a simple pattern recognition program that could recognize a variety of fasteners. Gaston and Lozano-Pérez [1983] have built a program

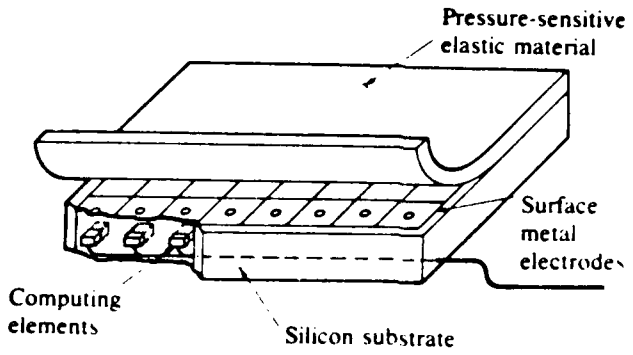


Figure 15. Architecture of the VLSI tactile sensor developed by Raibert and Tanner. A layer of pressure-sensitive rubber is placed in contact with a VLSI wafer. Metalization on the surface of the wafer forms large sensing electrodes that make contact with the pressure-sensitive rubber through holes in a protective layer of SiO_2, the overglass. (Reproduced from [Raibert and Tanner 1982, Figure 1])

that constructs an interpretation tree for a class of two-dimensional objects. The program assumes that there are n discrete sensors, at each of which the position and an approximate measure of the surface orientation is known. They show how two constraints, a distance constraint and a constraint on the normal directions at successive touch points, can substantially cut down the number of possible grasped object configurations. Grimson and Lozano-Pérez [1983] have extended the analysis to three dimensions. Faugeras and Hebert [1983] have developed a similar three-dimensional recognition and positioning algorithm using geometric matching between primitive surfaces. Bajcsy [1984] has investigated the use of two tactile sensors to determine the hardness and texture of surfaces.

6.3. Non-contact sensing

Non-contact sensing is important for a variety of applications in manufacturing. These include:

- *Inspection.* Most current industrial inspection uses binary two-dimensional images. Only recently have grey level systems become commercially available. No commercial system currently offers an modern edge detection system. Two dimensional inspection is appropriate for stamped or rotationally symmetric parts. Some experimental prototypes [Porter and Mundy 1982, Faugeras et. al. 1983] inspect surfaces such as engine mountings and airfoil blades.

- *parts acquisition.* Parts may be acquired from conveyor belts, from palettes, or from bins. Non-contact sensing means that the position of parts may not be accurately specified. Parts may have to be sorted if there is a possibility of more than one type being present.

- *determining grasp points.* Geometric analysis of shape allows grasp points to be determined [Brady 1982, Boissonat 1982].

Active sensing has been developed mainly for military applications. Image understanding is difficult and requires a great deal of computer power. FLIR, SAR, and millimeter radar imagery offer limited, computationally expedient, solutions to difficult vision problems. The algorithms that have been developed for isolating and identifying targets in natural scenes are restricted in scope. They do not generalize easily to manufacturing settings, where, for example, most objects are "hot".

Vision has the most highly developed theory, and the best sensors. Now one can get high quality solid state cameras. The rapid increase in the quality of solid state cameras has been accompanied by the rapid development of image understanding techniques.

Early vision processes include edge and region finding, texture analysis, and motion computation. All these operations are well suited to local parallel processing. Developments in edge finding include the work of Marr and Hildreth [1980], Haralick [1982], and Canny [1983]. Developments in grouping include the work of Lowe and Binford [1983]. Hildreth [1983] has developed a system for computing directional selectivity of motion using the Marr-Hildreth edge finder. Horn and Schunck [1981] and Schunck [1983] have shown how to compute the optic flow field from brightness patterns. (Bruss and Horn [1983] have developed an analysis of how the flow field can be used in passive navigation.) Brady [1983b, Brady 1983d, Brady and Asada 1984] has developed a new technique for representing two-dimensional shape, and has applied it to inspection.

The major breakthrough in vision over the past decade has been the development of three dimensional vision systems. These are usually referred to as "shape from" processes. Examples include: shape from stereo [Grimson 1981, Baker and Binford 1981, Binford 1984, Nishihara and Poggio 1984], shape from shading [Ikeuchi and Horn 1983], shape from contour [Witkin 1981, Brady and Yuille 1983], and shape from structured light [Porter and Mundy 1982, Faugeras 1983, Clocksin et. al. 1983, Bolles, Horaud, and Hannah 1984, Tsuji 1984].

Most of these "shape from" processes produce partial depth maps. Recently, fast techniques for interpolating full depth maps have been developed [Terzopoulos 1982, 1983]. A current area of intense investigation is the representation of surfaces [Faugeras, Hébert, and Ponce 1984, Shirai et. al. 1984, Ikeuchi and Horn 1983, Brady and Yuille 1984]. Finally, recent work by Brooks [1981] discusses object representation and the interaction between knowledge guided and data driven processing.

7. Reasoningthat connectsperceptionto action

This final section is speculative. It presents three examples of reasoning and problem solving that we are striving to make robots capable of. The aim is to illustrate the kinds of things we would like a robot to know, and the way in which

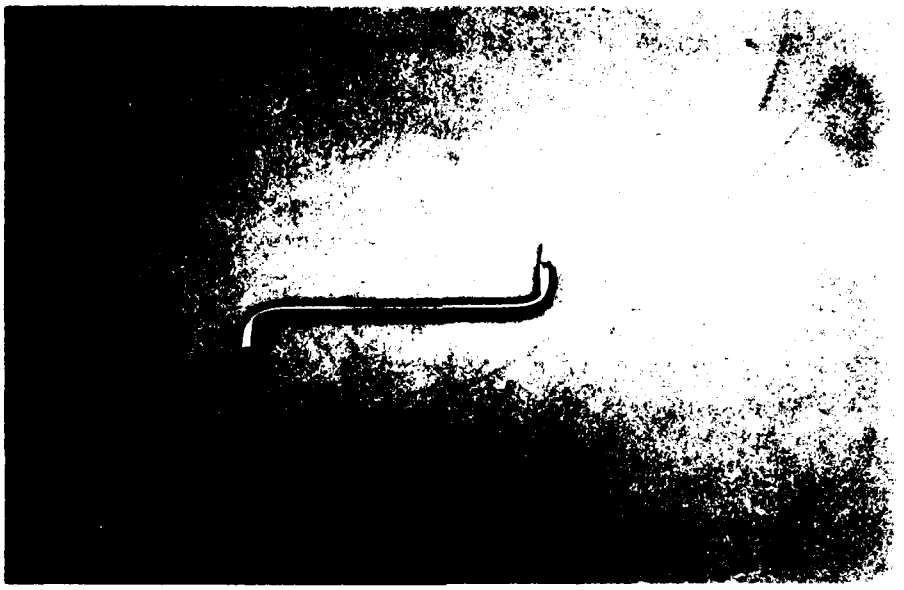


Figure 16. What is this tool, and what is it for?

that knowledge might be used. The knowledge that is involved concerns geometry, forces, process, space, and shape. The examples involve tools. They concern the interplay between the use or recognition of a tool and constraints on the use of tools. Reasoning between structure and function is particularly direct in the case of tools. Shape variations, though large (there are tens of kinds of hammer), are lessened by the fact that tools are rarely fussily adorned, since such adornments would get in the way of using the tool.

7.1. What is that tool for?

What is the tool illustrated in Figure 16, and how is it to be used? We (reasonably) suppose that a vision program [Brady and Asada 1984, Brady 1984] computes a description of the object that, based on the smoothed local symmetry axis, partially matches a *crank*. The model for a crank indicates that it is used by fixing the end P onto some object O, and rotating the object O about the symmetry axis at P by grasping the crank at the other end Q and rotating in a circle whose radius is the length of the horizontal arm of the crank. Further investigation of the crank model tells us that it is used for increasing the moment arm and hence the torque applied to the object O. We surmise that the tool is to be used for increasing torque on an object O. We have now decided (almost) how the tool is to be used, and we have a hypothesis about its purpose. The hypothesis is wrong.

The one thing that we do not yet know about how to use the tool is how to fix it at P to the object at O. There are many possibilities, the default being perhaps a socket connector for a nut (as for example on a tire lever). Closer inspection of the description computed by our vision program shows that the ends of the crank are screwdriver blades, set orthogonal to each other. Only screwdrivers (in our experience) have such blades. Apart from the blade, the tool bears some resemblance to a standard screwdriver, which also has a handle and a shaft. In

the standard screwdriver, however, the axes of the shaft and handle are collinear. Evidently, the tool is a special purpose screwdriver, since only screwdrivers have such blades.

Tools have the shape that they do in order to solve some problem that is difficult or impossible to solve with more generally useful forms. So why the crank shape? What problem is being solved that could not be solved with a more conventional screwdriver? Here are some screwdriver-specific instances of general problems that arise using tools:

- *Parts interface bug.* A part does not match the part to which it is being applied or fastened. For example, a wrench might be too small to span a nut; a sledgehammer is inappropriate for driving a tack. The screwdriver head may not match the screw (one might be Philips type). There is no evidence for this bug in Figure 16 because the fastener is not shown.

- *Restricted rotary motion bug.* A tool that is operated by turning it about some axis has encountered an obstruction that prevents it turning further. This bug occurs frequently in using wrenches. A socket wrench solves it by engaging a gear to turn the wrench in one direction, disengaging the gear to rotate in the other. How is it solved more generally? There is an analogous *restricted linear motion bug.* Think of an example! (one is given at the end of the section.)

- *Restricted access bug.* As anyone owning a particular (expensive) kind of British car knows only too well, often the hardest part of using a tool is mating it to the corresponding part. Many tools have an axis along, or about, which they are applied. The most common version of the restricted access bug is when the axis is too long to fit into the available space. In the case of screwdriving, this occurs when the screwdriver is restricted vertically above the screw. A short, stubby screwdriver is the usual solution to this problem.

Can the crank-screwdriver also overcome restricted access bugs? Of course. The geometric form of the crank-screwdriver is necessary to solve this restricted workspace problem, rather than being a torque magnifier as initially hypothesized. In fact, the tool is called an *offset screwdriver.* Figure 17 illustrates its use.

Since I first presented this example, another solution to the restricted access bug has been brought to my attention. Figure 18 shows a screwdriver whose shaft can bend about any axis orthogonal to it.

Why are the blades of an offset screwdriver set orthogonal to one another? Put differently, what bug do they help overcome? What would you need to know in order to figure it out?

No program is currently capable of the reasoning sketched above. Pieces of the required technology are available, admittedly in preliminary form, and there is cause for optimism that they could be made to work together appropriately. First, vision programs exist that can almost generate the necessary shape descriptions and model matching [Brady and Asada 1984, Brady 1984]. There is considerable interplay between form and function in the reasoning, and this has been initially explored by Winston, Binford, and their colleagues, combining the ACRONYM

Figure 17. An offset screwdriver overcomes the restricted access bug

Figure 18. A flexible screwdriver for solving restricted access bugs

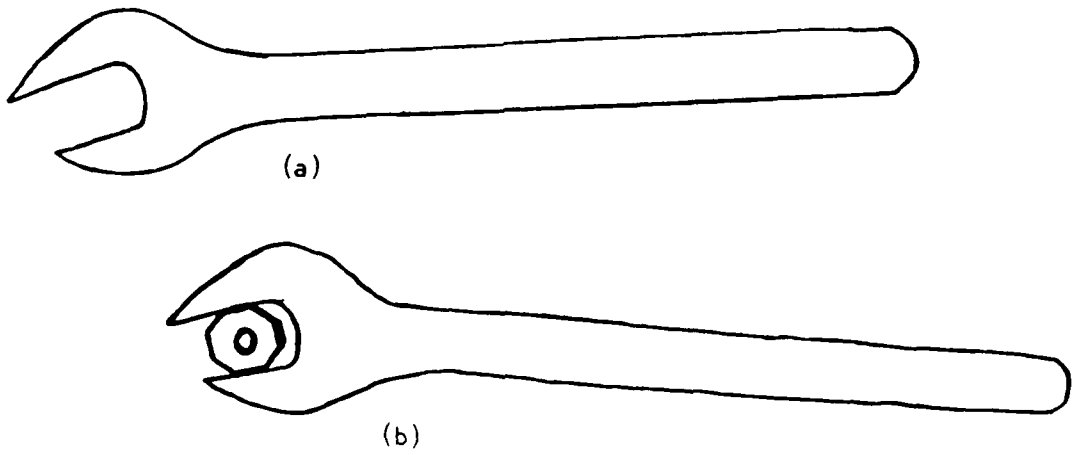


Figure 19. a. An asymmetric wrench. b. How to use a wrench.

system of shape description and Winston's analogy program [Winston, Binford, Katz, and Lowry 1982]. To figure out that the crucial thing about the form is its ability to overcome a restriction in the workspace, it is necessary to be able to reason about space and the swept volumes of objects. This is the contribution of Lozano-Pérez [1981, 1983], Brooks [1983b], and Lozano-Pérez, Mason, and Taylor [1983]. Forbus [1983] is developing a theory of processes, a system that can reason about physical processes like water flow, heat, and springs. This builds upon earlier work by DeKleer [1975] and Bundy [1979].

Answer to the problem: An example of a *restricted linear motion bug*: Trying to strike a nail with a hammer when there is insufficient space to swing the hammer.

7.2. Why are wrenches asymetric?

Figure 19a shows a standard (open-jawed) wrench. Why is it asymmetric? To understand this question, it is necessary to understand *how* it would most likely be judged asymmetric. This involves finding the head and handle [Brady and Asada 1984], assigning a "natural" coordinate frame to each [Brady 1982, 1984], and realizing that they do not line up. Since the handle is significantly longer that the head, it establishes a frame for the whole shape, so it is the head that is judged asymmetric about the handle frame.

Now that we at least understand the question, can we answer it? We are encouraged to relate a question of form to one of function. What is a wrench for, and how is it used? It is used as shown in Figure 19b: the head is placed against a nut; the handle is grasped and moved normal to its length; if the diameter of the nut and the opening of the jaws of the wrench match, the nut (assumed fixed) will cause the handle to rotate about the nut. Nowhere is mention made of asymmetry. Surely, a symmetric wrench would be easier to manufacture. Surely, a symmetric wrench would be equally good at turning nuts. Or would it?

all that questions of form often relate not just to function, but to solving problem that a "standard", here symmetric, wrench could not solve. The problem that arises using a wrench is the restricted rotary motion bug. In y tasks there is an interval $[o_1, o_2]$ (measured from the local coordinate frame responding to the axis of the handle) through which the wrench can be rotated. The crucial observation is that a wrench is (effectively) a lamina. As such, it has a degree of freedom corresponding to flipping it over. Exploiting this degree of freedom makes *no difference* to the effective workspace of a symmetric wrench. It *doubles* the workspace of an asymmetric wrench, giving $[-o_2, -o_1] \cup [o_1, o_2]$. In this way, an asymmetric wrench partially solves the restricted rotary motion bug. Perhaps this suggests how to *design* the head of a wrench, say by minimizing $[-o_1, o_1]$ subject to keeping the jaws parallel to each other. Perhaps it also suggests (for example to Winston's [1980] analogy program) that other turning tools should be asymmetric, for analogous reasons. There are many examples, of course, including the offset screwdriver discussed in the previous section.

7.3. How to disconnect a battery

The final example in this section concerns a familiar AI technique: debugging "almost right" plans. A battery is to be disconnected. The geometry of the terminal is shown in Figure 20a. A plan, devised previously, uses two socket wrenches, one acting as an adaptable fixture, the other to turn the bolt. The socket wrenches are applied along their axes, which coincide with the axes of the nut and bolt.

A new model of the battery-laden device is delivered. The plan will no longer work because there is an obstacle to the left of the head of the nut, restricting travel of the socket wrench along its axis. Using something akin to dependency directed reasoning, truth maintenance, or a similar technique for recording the causal connections in a chain of reasoning, we seek a method for adapting our almost-right plan to the new circumstance. There are a variety of techniques, one of which was illustrated in the offset screwdriver. That suggests bending a tool so that torque can be applied by pushing on a lever. In fact, socket wrenches have this feature built-in, so the problem was easy.

Unfortunately, the fix did not work. Figure 20b shows why. The new model has insufficient clearance for a socket wrench to fit around the head of the bolt. We further reconsider the plan, adding the new constraints, removing those parts that were dependent upon being able to use the socket wrench (none, in this case). The next idea is to use a different kind of wrench, but this will not work either, again because of the insufficient clearance. We need to be able to grasp one of the nut and bolt securely and turn the other. We can hold the bolt and turn the nut. Several ways are available, most obviously using a needlenose plier to secure the bolt.

8. Conclusion

Since Robotics is the connection of perception to action, Artificial Intelligence must have a central role in Robotics if the connection is to be intelligent. We have

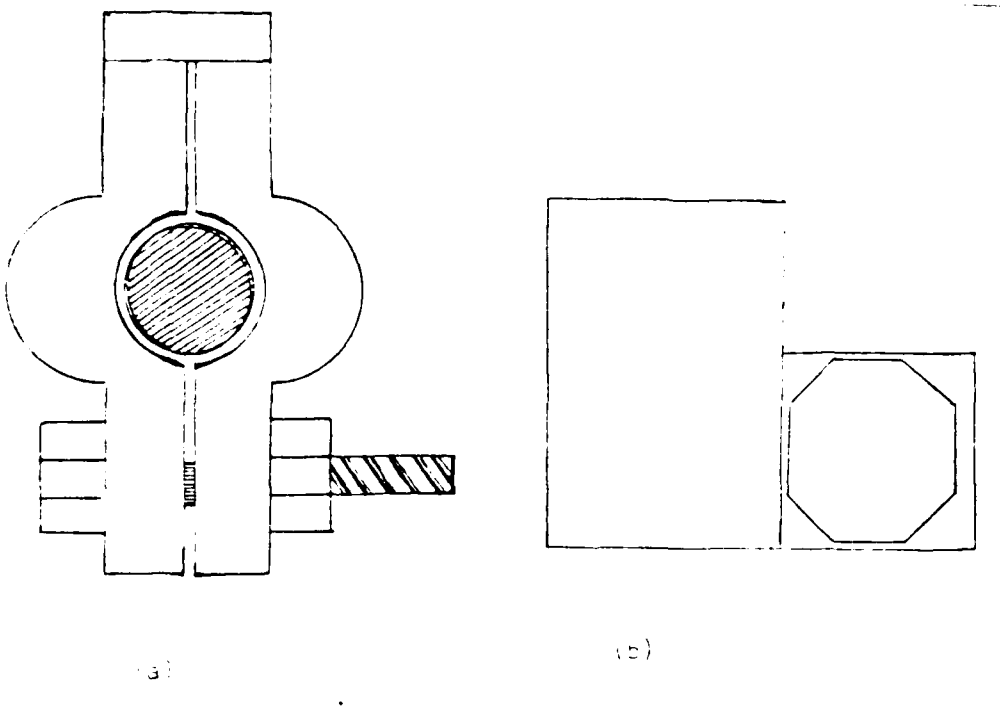


Figure 20. a. The geometry of a battery terminal. b. side view of the late model bolt.

illustrated current interactions between AI and Robotics and presented examples of the kinds of things we believe it is important for robots to know. We have discussed what robots should know, how that knowledge should be represented, and how it should be used. Robotics challenges AI by forcing it to deal with real objects in the real world. An important part of the challenge is dealing with rich geometric models.

9. Acknowledgements

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643, the Office of Naval Research under contract number N0014-80-C-0505, and the System Development Foundation. This paper is based on invited talks presented at the Annual Conference of the American Association for Artificial Intelligence, Washington, DC, August 1983, and the Quatrième Congrès des Reconnaissance des

Formes et d'Intelligence Artificielle, INRIA, Paris, January 1981. Many colleagues commented on earlier drafts or presentations of the material in this paper. I thank especially Phil Agre, Dave Braunegg, Bruce Donald, Olivier Faugeras, Georges Giralt, John Hollerbach, Dan Huttenlocher, Tomás Lozano-Pérez, Tommy Poggio, Marc Raibert, Ken Salisbury, Jon Taft, Dan Weld, and Patrick Winston.

10. References

Artificial Intelligence, [1984], "Special issue on: reasoning about structure and function," *Artificial Intelligence*, **22**.

Agin, G. J, [1980], "Computer vision systems for industrial inspection and assembly," *Computer*, **13**, 11-20.

Albus, J. S, [1984], "Integrated system control," .

Ambler, A.P., and R.J. Popplestone, [1975], "Inferring the positions of bodies from specified spatial relationships," *Artificial Intelligence*, **6**, 2, 157-174.

Asada, H, [June, 1982], "A characteristics analysis of manipulator dynamics using principal transformations," *Proc. Amer. Control Conf.*, Washington, D.C..

Asada, H., and Youcef-Toumi, K, [1984], Development of a direct-drive arm using high torque brushless motors, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Asada, H. and T. Kanade, [Aug. 1981], "Design concept of direct-drive manipulators using rare-earth DC torque motors," *Proc. 7th Int. Joint. Conf. Artificial Intelligence*, Vancouver, British Columbia, 775-778.

Bajcsy, R, [1984], What can we learn from one finger experiments?, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Baker H. Harlyn, and Binford T. O, [1981], "Depth from edge and intensity based stereo," *Int. Jt. Conf. Artif. Intell.*, **6**, .

Binford T. O, [1981], "Inferring surfaces from images," *Artificial Intelligence*, **17**, 205-245.

Binford, T. O, [1984], Stereo vision: complexity and constraints, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Boissonat, J. -D, [1982], "Stable matching between a hand structure and an object silhouette," *IEEE Patt. Anal. and Mach. Intell.*, **PAMI-4**, 603 - 611.

Brady, Michael, [1982], Parts description and acquisition using vision, *Robot vision* Rosenfeld, A (ed). Proc. SPIE, Washington D.C.,, 1 - 7.

Brady, Michael, [1983a], "Parallelism in vision," *Artificial Intelligence*, **21**, 271--284.

Brady, Michael, [1983b], Criteria for shape representations, *Human and Machine vision*, Beck J, and Rosenfeld A., eds., Academic Press.

Brady, Michael, [1983c], Trajectory planning, *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. J., Lozano-Pérez, T., and Mason, M. T., MIT Press, .

Brady, Michael, [1984], Representing shape, *Proc. IEEE Conference on Robotics*, Atlanta, Ga..

Brady, Michael, and Asada, Haruo, [1983], Smoothed local symmetries and their implementation, Proc. First Int. Symp. Robotics Research.

Brady, Michael, and Yuille, Alan, [1983], An extremum principle for shape from contour, MIT, AI Lab., MIT-AIM 711.

Brady, Michael, and Yuille, Alan, [1984], Representing three-dimensional shape, Romansy Conf., Udine, Italy.

Brooks, R.A, [1981], "Symbolic Reasoning Among 3-D Models and 2D Images," *Artificial Intelligence*, **17**, 285-348.

Brooks, R. A, [1982], "Symbolic error analysis and robot planning," *Int. Journal of Robotics Research*, **1 (4)**, 29 - 68.

Brooks, R. A, [1983a], "Solving the findpath problem by good representation of free space," *IEEE Trans. Sys. Man and Cyb.*, **SMC-13**, 190 - 197.

Brooks, R. A, [1983b], "Planning collision free motions for pick and place operations," *International Journal of Robotics Research*, **2 (4)**, .

Brooks, R. A., and Lozano-Pérez, Tomás, [1983], A subdivision algorithm in configuration space for findpath with rotation, Proc. Int. Jt. Conf. Artif. Intell. Karlsruhe.

Bruss A., and Horn, B. K. P, [1981], Passive Navigation, MIT, AI Memo 662.

Bundy, Alan, et. al, [1979], Solving mechanics problems using meta-level inference, *Expert systems in the microelectronic age*, Michie, D. (ed.), Edinburgh Univ. Press.

Cannon, R. H. Jr., and Schmitz, E, [1984], Precise control of flexible manipulators, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Canny, J. F, [1983 Sept], Finding lines and edges in images, *Proc. AAAI Conf.*, Washington, DC.

Canny, J. F, [1983], Finding lines and edges in images, MIT.

Clocksin, W. E., et. al, [1982], Progress in visual feedback for arc-welding of thin sheet steel, *Robot Vision*, Pugh, Alan ed., IFS.

Davis, Larry S. and Rosenfeld Azriel, [1981], "Cooperating processes for low-level vision: a survey," *Artificial Intelligence*, **17**, 245-265.

DeKleer, J, [1975], Qualitative and quantitative knowledge in classical mechanics, MIT Artificial Intelligence Laboratory, AI-TR-352.

Dobrotin, B. and Lewis, R, [1977], A Practical Manipulator System, Proc. Int. Jt. Conf. Artif. Intell. 5, Tblisi, USSR, 749-757.

Donald, Bruce R, [1983], Hypothesising channels through free space in solving the findpath problem, MIT Artificial Intelligence Laboratory, AIM-736.

Donald, Bruce R, [1984 (forthcoming)], Local And Global Techniques for Motion Planning, MIT Artificial Intelligence Laboratory.

Dubowsky, S. [1981], Model reference adaptive control, *NATO Advanced Study Institute on Robotics and Artificial Intelligence*, Brady, Michael and Gerhardt, Lester (eds), Springer Verlag.

Dubowsky, S., and DesForges, D. T. [1979], "The application of model-referenced adaptive control to robotic manipulators," *J. Dyn. Sys. Meas. Ctl.*, **101**, 193 – 200.

Dufay, B., and Latombe, J. -C. [1984], An approach to automatic robot programming based on inductive learning, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Engelberger, Joseph F, [1980], *Robotics in Practice*, Kogan Page.

Everett, H. R, [March/April, 1982], A Computer Controlled Sentry Robot: A Homebuilt Project Report, Robotics Age.

Faugeras, O., Hébert, M., Ponce, J., and Boissonat, J, [1982], Towards a flexible vision system, *Robot Vision*, Pugh, Alan ed., IFS.

Faugeras, O., and Hébert, M, [1983], A 3-D recognition and positioning algorithm using geometric matching between primitive surfaces, Proc. Int. Jt. Conf. Artif. Intell., Karlsruhe, 9996 – 1002.

Faugeras, O., Hébert, M., and Ponce, J, [1984], Object representation, identification, and positioning from range data, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Featherstone, R, [1983], "Position and velocity transformations between robot end effector coordinates and joint angles," *The International Journal of Robotics Research*, **2(2)**, .

Forbus, K. D, [1983], Qualitative process theory, MIT Artificial Intelligence Laboratory AIM-664A.

Franklin, James W., and VanderBrug, G. J, [March, 1982], Programming vision and robotics system with RAIL, Robots VI Conf., Detroit, SME.

Gaston, Peter C., and Lozano-Pérez, Tomás, [1983], Tactile recognition and localization using object models: the case of polyhedra on a plane, MIT Artificial Intelligence Lab. AIM-705.

Giralt, G, [1984], Mobile robots, *NATO Advanced Study Institute on Robotics and Artificial Intelligence*, Brady, Michael and Gerhardt, Lester (eds), Springer Verlag.

Giralt, G., Sobek, R. and Chatila, R, [1977], A Muti-Level Planning and Navigation System for a Mobile Robot; A First Approach to HILARE, Proc. Int. Jt. Conf. Artif. Intell. 6, MIT.

Giralt, G., Chatila, R., and Vaisset, M, [1984], An integrated navigation and motion control system for autonomous multisensory mobile robots, *Proceedings*

of the International Symposium on Robotics Research. 1 Michael Brady and Richard Paul (eds.), MIT Press.

Hillis, D, [1982], "A high-resolution image touch sensor," *Int. J. Robotics Res.*, **1(2)**.

Goto, T., K. Takeyasu, and T. Inoyama, [1980], "Control algorithm for precision insert operation robots," *IEEE Trans. Systems, Man, Cybernetics*, SMC-10, 1, 19 25.

Grimson, W. E. L, [1981], *From images to surfaces: a computational study of the human early visual system*, MIT Press, Cambridge.

Grimson, W. E. L, and Lozano-Pérez, T, [1984], "Local constraints in tactile recognition," *International Journal of Robotics Research*, **3 (3)**.

Hackwood, S., and Beni , [1983], "Torque sensitive tactile array for robotics," *Int. Jour. Robotics Research*, **2 (2)**, .

Hanafusa, H., and Asada, H, [1977], Stable prehension by a robot hand with elastic fingers, Int. Symp. Ind. Rob. 7, 361 — 368.

Hanafusa, H., and Asada, H, [1979], A robot hand with elastic fingers and its application to assembly process, IFAC Symp. Inf. Ctl., Problems in Manuf. Tech., 127 — 138.

Haralick, Robert M., Watson, Layne T., and Laffey, Thomas J, [1983], "The topographic primal sketch," *The International Journal of Robotics Research*, **2 (1)**, 50 - 72.

Harmon L, [1982], "Automated Tactile Sensing," *Int. Jour. Robotics Research*, **1 (2)**, 3 - 33.

Harmon L, [1982], "Robotic taction for industrial assembly," *Int. Jour. Robotics Research*, **3 (1)**.

Harmon S. Y, [May, 1983], Coordination Between Control and Knowledge Based Systems for Autonomous Vehicle Guidance, Proc. IEEE Trends and Applications Conference, Gaithersburg, MD.

Harmon, S. Y, [April, 1983], Information Processing System Architecture for an Autonomous Robot System, Conference on Artificial Intelligence, Oakland University, Rochester, MI.

Hayes-Roth, P., Waterman, D., and Lenat. D. (eds)., [1983], *Building expert systems*, Addison Wesley.

Hildreth, E, [1983], The measurement of visual motion, MIT, Artificial Intelligence Laboratory.

Hillis, W. Daniel, [1982], "A high-resolution image touch sensor," *Int. Jour. Robotics Research*, **1 (2)**, 33 - 44.

Hirose, S., Nose, M., Kikuchi, H., and Umetani, Y, [1984], Adaptive gait control of a quadruped walking vehicle, *Proceedings of the International*

Symposium on Robotics Research: 1 Michael Brady and Richard Paul (eds.), MIT Press.

Hobbs, J., and Moore, R, [1984], *Formal theories of the common sense world*, Ablex.

Hollerbach, J. M, [1983], Dynamics, *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. L., Lozano Pérez, T., and Mason, M. T., MIT Press, .

Hollerbach, J. M., and Sahar, Gideon, [1983], Wrist partitioned inverse kinematic accelerations and manipulator dynamics, MIT Artificial Intelligence Laboratory, AIM-717.

Hopcroft, J. E., Schwartz, J. T., and Sharir M, [1983], "Efficient detection of intersections among spheres," *The International Journal of Robotics Research*, **2 (4)**.

Horn B. K. P, [1982], Sequins and Quills - Representations for Surface Topography, *Representation of 3-Dimensional Objects* ed. Bajcsy R., Springer Verlag.

Horn and Schunck, [1982], "Determining Optical Flow," *Artificial Intelligence*, **17**, 185 - 203.

Huston, Ronald L., and Kelly, Frederick A, [1982], "The development of equations of motion of single-arm robots," *IEEE Sys. Man and Cyb.*, **SMC-12**, 259 -- 266.

Ikeuchi K, [1981], "Determination of surface orientations of specular surfaces by using the photometric stereo method," *IEEE*, , .

Ikeuchi, K, [1981], Recognition of 3D object using extended Gaussian image, Proc. Int. Jt. Conf. Artif. Intell. 7.

Ikeuchi K. and Horn B. K. P, [1981], "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, **17**, 141-185.

Ikeuchi, K., Horn, B. K. P., Nagata, S., and Callahan, T, [1984], Picking up an object from a pile of objects, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Jacobsen, S. C., et. al, [1984], The Utah/MIT dextrous hand -- work in progress, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Kahn, M. E., and Roth, B, [1971], "The near minimum-time control of open-loop articulated kinematic chains," *J. Dyn. Sys. Meas. Ctl.*, **93**, 164 -- 172.

Kanade, T., and Sommer, T, [1984], An optical proximity sensor for measuring surface position and orientation for robot manipulation, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Kane, Thomas R., and Levinson, David A, [1983], "The use of Kane's dynamical equations in Robotics," *Int. J. Rob. Res.*, **2 no. 3**, 3 -- 22.

Klein C. A., and Wahawisan, J. J, [1982], "Use of a multiprocessor for control of a robotic system," *Int. J. Rob. Res.*, **1 no. 2**, 45 59.

Klein, C. A., Olson, K. W., and Pugh, D. R, [1983], "Use of force and attitude sensors for locomotion of a legged vehicle over irregular terrain," *Int. J. Rob. Res.*, **2 no. 2**, 3 17.

Lewis, R. A. and Bejczy, A. K, [1973], Planning Considerations For A Roving Robot With Arm, Proc. Int. Jt. Conf. Artif. Intell. 3, Stanford University, 308-315.

Lewis, R. A. and Johnston, A. R, [1977], A Scanning Laser Rangefinder For A Roving Robot With Arm, Proc. Int. Jt. Conf. Artif. Intell. 5, Tblisi, USSR, 762-768.

Lieberman, L.I., and M.A. Wesley, [1977], "AUTOPASS: an automatic programming system for computer controlled mechanical assembly," *IBM J. Research Development*, **21**, 4, 321-333.

Lowe, D. G., and Binford, T. O, [1982], Segmentation and aggregation: an approach to figure-ground phenomena, Proc. Image Understanding Workshop, Baumann Lee S. (ed.), Sci. App. Inc. Tysons Corner Va., 168 - 178.

Lozano-Pérez, T, [1976], The design of a mechanical assembly system, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI TR 397.

Lozano-Pérez, Tomás l, [1981], "Automatic planning of manipulator transfer movements," *IEEE Trans. Sys., Man and Cyb.*, **SMC-11**, 681-698.

Lozano-Pérez, Tomás l, [1983a], "Spatial planning: a configuration space approach," *IEEE Trans. Comp.*, **C-32**, 108-120.

Lozano-Pérez, Tomás, [1983b], "Robot programming," *Proc. IEEE*, **71**, 821 - 841.

Lozano-Pérez, Tomás, [1983c], Spatial Reasoning, *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. J., Lozano-Pérez, T., and Mason, M. T., MIT Press, .

Lozano-Pérez, Tomás, Mason, Matthew T., and Taylor, R. H, [1984], Automatic Synthesis of fine-motion strategies for robots, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Marr, D, [1982], *Vision*, Freeman, San Francisco.

Marr, D. and Hildreth, E.C, [1980], "Theory of Edge Detection," *Proc. R. Soc. Lond. B*, **270**, 187-217.

Marr D. and Poggio T, [1979], "A theory of human stereo vision," *Proc. R. Soc. Lond. B*, **204**, 301-328.

Mason, T. M,(reprinted in *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. J., Lozano-Pérez, T., and Mason, M. T., MIT Press) [1981], "Compliance and force control for computer controlled manipulators," *IEEE Trans. Sys. Man and Cyb.*, **SMC-11**, 418 - 432.

Mason, T. M, [1983]. Compliance, *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. J., Lozano-Pérez, T., and Mason, M. T., MIT Press, .

Mason, T. M, [1982]. Manipulator grasping and pushing operations, MIT.

Michie, D, [1979]. *Expert systems in the microelectronic age*, Ellis Horwood.

Moravec, Hans P, [1981]. *Robot rover visual navigation*, UMI Research Press, Ann Arbor, MI.

Moravec, Hans P, [July, 1983], "The Stanford cart and the CMU rover," *IEEE Trans. Ind. Electronics.*

Moravec, Hans P, [1984], Locomotion, vision, and intelligence, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Nakagawa, Y., and Ninomiya, T, [1984], Structured light method for inspection of solder joints and assembly robot vision system, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Nilsson N. J, [1969], A mobile automaton: an application of Artificial Intelligence techniques, Proc. Int. Jt. Conf. Artif. Intell. 1, Washington, DC.

Nishihara, H. K., and Poggio, T, [1984], Stereo vision for robotics, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Ohwovoriole, M. S., and Roth, B, [1981], "An extension of screw theory," *Trans. ASME, J. of Mech. Design*, **103**, 725 – 735.

Okada, T, [1979], Computer control of multi-jointed finger system, Int. Jt. COnf. Artif. Intell. 6, Tokyo.

Okada, T, [1982], "Development of an optical distance sensor for robots," *Int. J. Rob. Res.*, **1 no. 4**, 3 – 14.

Orin, D. E., and Schrader, W. W, [1984], Efficient Jacobian determination for robot manipulators, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Ozguner, F., Tsai, S. J., and McGhee, R. B, [1984], Rough terrain locomotion by a hexapod robot using a binocular ranging system, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Paul, R.P, [1981], *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, Mass..

Paul, R. C., and Shimano, B. E, [1976], Compliance and control, Proc. Jt. Aut. Control Conf., 694 – 699.

Paul, R. C., Stevenson, C. N., and Renaud, M, [1984], A systematic approach for obtaining the kinematics of recursive manipulators based on homogeneous

transformations, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Pieper, D.L, [1968], The Kinematics of Manipulators under Computer Control, Ph.D. thesis, Department of Computer Science, Stanford University.

Pieper, D.L., and B. Roth, [September 1969], "The kinematics of manipulators under computer control," *Proc. 2nd Int. Conf. Theory of Machines and Mechanisms*, Warsaw.

Popplestone, R. J., Ambler, A. P., and Bellos, I. M, [1980], "An interpreter for a language for describing assemblies," *Artificial Intelligence*, **14**, 79-107.

Porter, G., and Mundy, J, [1982], "A non-contact profile sensor system for visual inspections," *IEEE Workshop on Ind. Appl. of Mach. Vis.*, , .

Porter, G., and Mundy, J, [1984], A model-driven visual inspection module, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Raibert, M. H (Guest editor), [1984], "Special Issue on Walking machines," *Int. J. Robotics Research*, **3, no. 2**.

Raibert, M. H., and Craig, J. J, [1983], A hybrid force and position controller, *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. J., Lozano-Pérez, T., and Mason, M. T., MIT Press, .

Raibert, Marc H., and Tanner, John E, [1982], "Design and implementation of a VLSI tactile sensing computer," *Int. Jour. Robotics Research*, **1 (3)**, 3 – 18.

Raibert, M. H., and Sutherland, I. E, [1983], "Machines that walk," *Scientific American*, **248**, 44 — 53.

Raibert, M. H., Brown, H. B. Jr., and Murthy, S. S, [1984], 3D balance using 2D algorithms?, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Renaud, M, [1984], An efficient iterative analytical procedure for obtaining a robot manipulator dynamic model, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Requicha, A. A. G, [December, 1980], "Representation of Rigid Solids: Theory, Methods, and Systems," *Computing Surveys*, **12**, 4, 437–464.

Rich, C. R., and Waters, R, [1981], Abstraction, inspection, and debugging in programming, MIT Artificial Intelligence Laboratory, AIM-634.

Roth, B, [1984], Screws, motors, and wrenches that cannot be bought in a hardware store, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Sacerdoti, E, [1975], A structure for plans and behavior, SRI Artificial Intelligence Center TR-109.

Salisbury, J.K, [1982], Kinematic and Force Analysis of Articulated Hands. Ph D thesis, Department of Mechanical Engineering, Stanford University.

Salisbury, J. K, [November 1980], Active Stiffness Control of a Manipulator in Cartesian Coordinates, IEEE Conference on Decision and Control, Albuquerque, New Mexico.

Salisbury, J. K, [1982], Kinematic and force analysis of articulated hands, Stanford University.

Salisbury, J. K, [1984], Interpretation of contact geometries from force measurements, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Salisbury, J.K., and J.J. Craig, [1982], "Articulated hands: force control and kinematic issues," *Int. J. Robotics Research*, **1**, 1, 4-17.

Schunck, B. G, [1983], Motion segmentation and estimation, MIT Artificial Intelligence Laboratory.

Schwartz, Jacob T., and Sharir, Micha, [1983], "The piano movers problem III," *The International Journal of Robotics Research*, **2 (3)**.

Shimano, B. E., Geschke, C. C., and Spaulding, C. H, [1984], VAL II: a robot programming language and control system, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Shirai, Y., Koshikawa, K., Oshima, M., and Ikeuchi, K, [1984], An approach to object recognition using 3-D solid model, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Taylor, R.H, [July, 1976], The synthesis of manipulator control programs from task-level specifications, Artificial Intelligence Laboratory, Stanford University, AIM-282.

Taylor, R. H., Summers, P. D., and Meyer J. M, [1982], "AML: a manufacturing language," *The International Journal of Robotics Research*, **1(3)**, 19 - 41.

Terzopoulos, D., [1982], Multilevel reconstruction of visual surfaces: Variational principles and finite element representations, MIT A.I. Lab., Cambridge, MA, 1982, AI Memo No. 671, to appear in *Multiresolution Image Processing and Analysis*, A. Rosenfeld (ed.), Springer Verlag, New York, 1984.

Terzopoulos, D., [1983], "Multilevel computational processes for visual surface reconstruction," *Computer Vision, Graphics, and Image Processing*, **24**, 52-96.

Trevelyan, J. P., Kovesi, P. D., and Ong, M. C. H, [1984], Motion control for a sheep shearing robot, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Tsuji, S., and Asada, M, [1984], Understanding of three-dimensional motion in time-varying imagery, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

VAL, [1980], User's guide: a robot programming and control system, CONDEC Unimation Robotics.

Villers, Philippe, [1982], Present industrial use of vision sensors for robot guidance, *Robot Vision*, Pugh, Alan ed., IFS.

Vilnrotter F., Nevatia R., and Price K. E, [1981], Structural analysis of natural textures, *Proc. Image Understanding Workshop* ed. Lee Baumann S., 61-68.

Wesley, M. A. et al, [January, 1980], "A Geometric Modeling System for Automated Mechanical Assembly," *IBM J. Research and Development*, **24**, 1, 64 74.

Whitney, D. E, [1983], The mathematics of compliance, *Robot motion: planning and control*, Brady, Michael, Hollerbach, J. M., Johnson, T. J., Lozano-Pérez, T., and Mason, M. T., MIT Press, .

Winston, Patrick H, [1980], "Learning and reasoning by analogy," *Comm. ACM*, **23**.

Winston, Patrick H, [1984], *Artificial Intelligence, Second Edition*, Addison Wesley, Reading: Mass..

Winston, Patrick H., Binford, Thomas O., Katz, Boris, and Lowry, Michael, [1984], Learning physical descriptions from functional descriptions, examples, and precedents, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Witkin, Andrew P, [1981], "Recovering surface shape and orientation from texture," *Artificial Intelligence*, **17**, 17-47.

Woodham, Robert J, [1981], "Analysing images of curved surfaces," *Artificial Intelligence*, **17**, 117 — 140.

Yoshikawa, T, [1984], Analysis and control of robot manipulators with redundancy, *Proceedings of the International Symposium on Robotics Research: 1* Michael Brady and Richard Paul (eds.), MIT Press.

Young, K. K. D, [1978], "Controller design for a manipulator using theory of variable structure systems," *IEEE Trans. Sys. Man and Cyb.*, **SMC-8**, 101 — 109.

Zucker S. W., Hummel R. A., and Rosenfeld Azriel, [1977], "An application of relaxation labelling to line and curve enhancement," *IEEE Trans. Computers*, **C-26**, 394-403, 922-929.

FILMED

8