THE ACQUISITION, PROCESSING, AND USE OF TACTILE SENSOR DATA IN ROBOT CONTROL

Kenneth J. Overton

COINS Technical Report 84-08

May 1984

Laboratory for Perceptual Robotics
Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

This document was supported by the National Science Foundation under Grant ECS-8108818.

The following are trademarks of Digital Equipment Corporation:
DEC PDP VAX VMS DIGITAL

ANSYS is a trademark of Swanson Analysis Systems, Inc.

SC-Consil and Consil-II are trademarks of Technical Wire Products, Inc.

Tecknit is a trademark of the Tecknit company.

National Science Foundation
ECS-8108818

This work is dedicated to

Charlotte and A.J.

for the support they have provided
during a long academic career.

Acknowledgments

Doctoral dissertations in the field of Computer Science require an abundance of resources, primarily in the form of computers. Robotics, as a field intersecting with Computer Science, requires not only these resources but also very specific devices of its own in the form of manipulators, sensors, etc. This work would not have been possible without the support, both intellectual and in the form of hardware, provided by a number of individuals and groups. I would like to take this opportunity to express my appreciation for the help and encouragement provided by these people.

To begin with I would like to thank Professors Michael Arbib, Ed Riseman, and Paul Herron, who formed my Ph.D. committee, for their encouragement and intellectual stimulation -- and for the time spent pushing me to finish. In particular I wish to thank Professor Michael Arbib for agreeing to chair my committee in an area which was new to the department. His support and enthusiasm has contributed greatly not only to this work but to the establishment of a continuing research facility in the Computer and Information Science Department (COINS).

The Corporate Research and Architecture Group (CRA) of Digital Equipment Corporation (DEC) also has my sincere thanks. Dr. Tom Williams, manager of the Manufacturing Automation Program of DEC's CRA group has supported this work from the very early stages and will always remain a colleague and close friend. Acknowledgment needs to go to Drs. Sam Fuller and Bob Glorioso, the present and past managers of the CRA Group, who have continued to support both my work at DEC as well as the automation research program in general. The foresight shown by these individuals is of benefit not only to Digital as a company but also to the various areas of research in computer science and robotics. A word of acknowledgment goes to Gerry Roston for his help with the finite element modelling. Special thanks also goes to Karen for the long evenings and weekends spent helping with the final preparation of this document. As a final note regarding DEC, I wish to thank the members of the Manufacturing Automation Program, in particular Tom and Leo, for providing an enjoyable environment in which to work.

As as outgrowth of this work and my collaboration with Michael Arbib, a laboratory for robotic research has come into existence in the COINS Department at the University of Massachusetts. The initiation of a new laboratory is always a difficult and time consuming proposition. The Laboratory for Perceptual Robotics (LPR) would not now exist but for the assistance of a large number of individuals. I would like to thank Ed Riseman and Michael Arbib for their continuing conceptual guidance and political support of the research. I have interacted with a number of bright and interesting people in the lab and it is these people who have forced the lab to succeed. Without the contributions of Judy, Gerry, Terri, Randy, Damion, TV, Pradip, and Steve the project would not have the character or scope it now exhibits.

A final note of appreciation goes to Dr. Norman Caplan, program director

of the Automation, Bioengineering and Sensing Systems Program for the National Science Foundation for providing grant number ECS-8108818 which formed the basis for the LPR.

Abstract

The Acquisition, Processing, and Use of Tactile Sensor Data in Robot Control

May 1984

Kenneth Jackson Overton, B.A., Indiana University
M.S., University of Massachusetts, Ph.D., University of Massachusetts
Directed by: Professor Michael Arbib

Robots are machines capable of interacting with their environments in intelligent manners. In order for such capabilities to exist, these machines must be able to act relative to their environments, sense quantities about both themselves and their environments, and make decisions guided by the sensor input. This dissertation is directed toward developing a sense of touch for robot systems. Robot senses are discussed and a classification scheme developed with particular attention paid to the definition of tactile sensors. Current tactile sensors in use in laboratories are presented along with the design of the sensor developed in this work. Issues surrounding the response characteristics of the sensor and static and dynamic tactile image processing are presented and several experiments discussed. An approach to the utilization of sensory information in high-level control provides the topic for the chapter on schemas. Conclusions drawn from this work as well as directions for future research in tactile sensation are provided in the final section and appendices.

Table of Contents

List of Tables

List of Figures

Introduction

Humans function in a very complex physical world. This complexity is dealt with by combining sensory information, logical thought processes, and the memories of past situations to formulate appropriate behavior. Machines, including robots, are being used increasingly in situations previously reserved for humans. The acquisition, processing, and use of sensory information by humans performing delicate manipulative tasks is quite complex compared to the capabilities of current robots. In order to develop more sophisticated robots, research is needed on the increasingly subtle use of sensory input in their control. This dissertation presents experiments in the processing and use of tactile array sensor information for the guidance of robot arms and hands.

The field of robotics is in rapid development as robots become more numerous, adaptable, reliable, precise, and inexpensive. They are making increasing use of sensory feedback about their relation to their work environments. Current robots, ranging from the commercially available to experimental prototypes, are beginning to be equipped with optical, tactile, force, and a variety of special purpose sensors. (For an introduction to robotics, the reader is refered to [176] and [60].) This makes for robots which are able to react to the particulars of a dynamic environment in a flexible way. (A major focus in the field of robotics is the area of computer controlled robot assembly. For detailed discussions of the problems, sensors and approaches, the reader is referred to [211], [87], [181], [145], [25], [57], [54], [55], [143], [144], [147] and [214]. In addition, the robot has become the vehicle for research into the development of machine perception and machine intelligence in various real situations.

Robotics development also has considerable significance to United States industry, see [60] for a more complete discussion. The use of sensory feedback will extend the range of behaviors for robots beyond the simple pick-and-place operations. Consider the difficulties of knotting wires, placing pins in holes, or using a screwdriver without force feedback. Increased intelligence will lessen the need for organizing the manufacturing process around the limitations of the robot system. A manufacturing cell could employ a semi-autonomous robot with only a small selection of jigs, feeders, and other attachments for the positioning of objects. Sophisticated, small batch manufacturing could then be reconfigured to meet changing demands through software level programming of the robots without extensive retooling of the work environment. This would directly lead to the increases in productivity and reliability of products required to remain competitive with foreign industrial concerns [145], [204].

The sensory capabilities of present robots, however, are quite primitive and the typical industrial robot today is, by human standards, "blind", "deaf", "dumb", and cannot "feel". In addition to a lack of sensory capabilities, the work to date has exhibited a distinct task-by-task approach with little regard for a general theory for the integration of sensory information into the control of robots. In most cases, the task is completely specified and is performed in a heavily constrained, known environment, e.g. see [180]. Recently, vision systems

and certain forms of force compensation have found increasing use in helping to remove the requirement of complete specificity, e.g. [179]. A more complete discussion of machine vision is provided in Chapter 2. The state of the art in robotics, however, has yet to include areas such as tactile sensation and dynamic visual input, flexible control structures, and data-driven models of the environment. Research in these areas, among others [26], is needed to bring robotics to a more universally applicable state.

An area receiving limited attention in the literature but comprising the fundamental component of this work involves the of use of tactile sensory data in the control of robot systems. Research in this area is ongoing and for further discussions, the reader is referred to [193], [119], [104], [53], [98], [93], [94]. (For a review, see [95].) This research in the use of tactile data by robot manipulators has been limited mainly to gross force sensing in the wrist, and in some cases the fingers, for two general reasons. First, until recently, there has been a lack of high resolution sensing arrays which could be utilized in the end effectors of a manipulator. Second, and probably more important, people are far less conscious of the role that the "sense of touch" plays in everyday activity, as compared with vision. While vision may continue to be the primary sense utilized in robots, tactile sensing fills an important role, not only when vision is inapplicable due to inherent ambiguities, but in the very fine control of delicate manipulation. It must, therefore, be understood fully.

The areas considered in this work include: static and dynamic tactile sensation, and the interactions of multimodal sensory data in the creation and maintenance of environmental models (schema-assemblages). The approach includes development of a tactile array sensor, development of both static and dynamic force image processing techniques, and experiments in control based on the concept of a schema. The methods are both theoretical and experimental. From the theoretical aspect, general principles are sought and algorithms developed. These ideas are then formulated into computer programs and studied when applied to tactile images. The experimental stage consists of exploring the behavior of processing algorithms and a robot gripper equipped with tactile array sensor.

The constraints and characteristics of the tactile domain are discussed in relation to the effects they have on algorithm development. A basic premise throughout this work contends that a great deal of useful information is available from tactile array sensor data from relatively simple processing algorithms which make use of the constraints of the domain. Both static and dynamic tactile data processing algorithms are developed.

This work investigates what may be termed robot "passive touch." This includes, but goes beyond, pattern recognition from single static force images. There are certain conceptual parallels between the "force images" used here and the digitized images of a natural scene as studied by machine vision research groups. In both cases one can have an underlying model of an object in the environment whose position and orientation is being sought. In the case of a perfect, static model, the desired output of a "force image" understanding /

analysis system is the location of the sensor on the surface of the object and, therefore, the object's relative orientation and location in space.

The ability to process sensory data is necessary; but in order to fully utilize the resulting information, it must be integrated into the control of the robot. The control structure may be conceptually divided into two components. The first is the structure which actually controls the motor behavior of the robot. This system is responsible for manipulating the information about the world, processing the sensory data, and establishing the actions of the machine. The other component consists of some form of representation of the objects in the world, their relationships to one another, the representation of the sensory input, and mechanisms for relating the various sensory modalities to one another and to the objects in the world, cf. [47], [136], [128], [85].

This work analyzes the role of the following data structures and processes in the sensory control of a robot: the representation of the relationships of the objects in the environment to the robot, the updating of that representation as the robot moves and as new sensory data are gathered, and the use of the representation by programs which control the movement, these forming a cycle of integrated perception and action. We seek to understand the design of units which cooperate in achieving multisensory coordination. The analysis seeks to decompose functions into the interaction of a family of simultaneously active processes called schemas [11], [12], [13], [14], which will serve as building blocks for both representations and programs. (For a discussion of schemas as they have been used in AI, see [28] and [135].)

This thesis is divided into six chapters plus introductory and conclusion sections. Chapter 1 discusses robot senses and presents a taxonomy for the classification of sensors. Chapter 2 compares and contrasts the tactile array sensors in use and under development in research labs in this country and in Europe. A detailed analysis of the development of the array sensor used in this work is also provided in Chapter 2. The response characteristics for the device and a discussion of a number of tactile image correction algorithms is contained in Chapter 3. Static tactile image processing is the topic of Chapter 4 with the issues surrounding dynamic tactile image processing addressed in Chapter 5. In both cases the characteristics of the domain are discussed and used to simplify the algorithms employed to extract a given piece of information. Chapter 6 is devoted to the development of a schema approach to the utilization of sensory information in control. The approach taken is to develop an interaction-based, distributed control structure which implements the idea of coordinated sensing and acting. Conclusions and a discussion of directions for further work comprises the final section. The appendices include discussions of a communication link built to connect a DEC MINC system and an IBM RS1 manipulator system, possible extensions of the sensor design to other devices, and a set of psycho-physical experiments designed to provide some insight into the connection between the tactile sensory experience and semantic labels.

Chapter 1

Robot Senses

1.1 Introduction

The topic of this thesis is the acquisition, processing, and use of tactile, or touch, sensor data in the manipulation of objects by machines. In order to understand the relationships between the tactile domain and other sensor systems, one must be aware of the characteristics of those other domains. This chapter provides an overview of the various sensing modalities and the associated sensors available to robots. A classification system is developed and presented as a possible framework in which to discuss robotic sensory systems. Greater emphasis is placed on the tactile domain than other domains since it is the topic of the thesis.

A detailed discussion and analysis of current tactile array sensor designs is given in Chapter 2 with specific emphasis placed on the relationships between these sensors and the one developed during for this work.

1.2 A Taxonomy for Robot Sensory Information

1.2.1 Taxonomy: Part 1

One possible classification characteristic is the sensing modality employed by the device. Sensation may be divided into several general classes known as modalities. In the biological domain, these include sight, hearing, taste, touch, etc. Submodalities are specific areas within a modality, e.g. temperature and pain within touch. The extent of physical interaction between the sensor and the object whose qualities are being transduced may also be used as a classification parameter. Another distinguishing factor involves the physical quantity transduced or information provided by the sensor. In general, there are several types of sensors which transduce the same physical quantity. For example, the distance between the robot and an object in its environment could be determined through the use of stereo vision in the visible spectrum, laser rangefinding, or acoustical techniques. The sensors corresponding to these three techniques differ greatly in their design and the underlying physics of their operation yet can provide the same information to the robot.

Sensory information may be broken into two general classes: that which is obtained *DIRECTLY* by the robot and that which is obtained *INDIRECTLY*, see Figure 1.1. *DIRECT* sensory information is provided by sensors/systems local to the robot and may include visual and tactile information. *INDIRECT* information is provided to the machine by external systems and could include information regarding the relative positions of other machines which are out of the range of the robot's sensing systems. The *DIRECT* vs. *INDIRECT* classification deals more with the source of the information than with its content since most, if not all, forms of information can either be gathered by the robot's systems or supplied to the robot from an outside source. The purpose of this distinction is to differentiate between information which is obtained by sensors local to the robot

system and that which is obtained remotely. Thus when considering sensing systems for robots, only those sensors providing *DIRECT* information are considered. In practice, the factors determining whether a specific sensor is located on (in) each robot system (*DIRECT*) or whether they are located external to the robot (*INDIRECT*) are usually the cost of the sensor, the amount of processing required, the proximity of the robot to external sensing systems, and the bandwidth of the data path between the robot and the external system. (Information regarding the velocity of a Mars rover must be available directly to the rover as opposed to having this information transmitted from an Earth station.) Of interest in the present work are the sensors which are directly usable by the robot system.

Sensory information may be further categorized according to whether it concerns events or quantities which are *INTERNAL* or *EXTERNAL* to the robot. As in biological systems, *INTERNAL* sensing provides information regarding the state of the organism itself: quantities such as the forces present at various joints, the velocities of the limbs, temperature of the organism, and the status of various subsystems, e.g. the circulation system. Analogous quantities are available to robots, with the temperatures of interest being those of the various motors and circuits and with subsystems including the bus(es), processors, drive systems, etc. *EXTERNAL* information concerns the environment in which the robot exists and its relation to the robot, e.g. the locations and orientations of objects and the motion of the robot relative to the environment.

The final two levels in the hierarchy presented in Figure 1.1 discuss the spatial and temporal extents of the characteristic transduced. Since this work deals primarily with tactile sensation, motivation for these distinctions will be drawn from the human cutaneous sensing system.

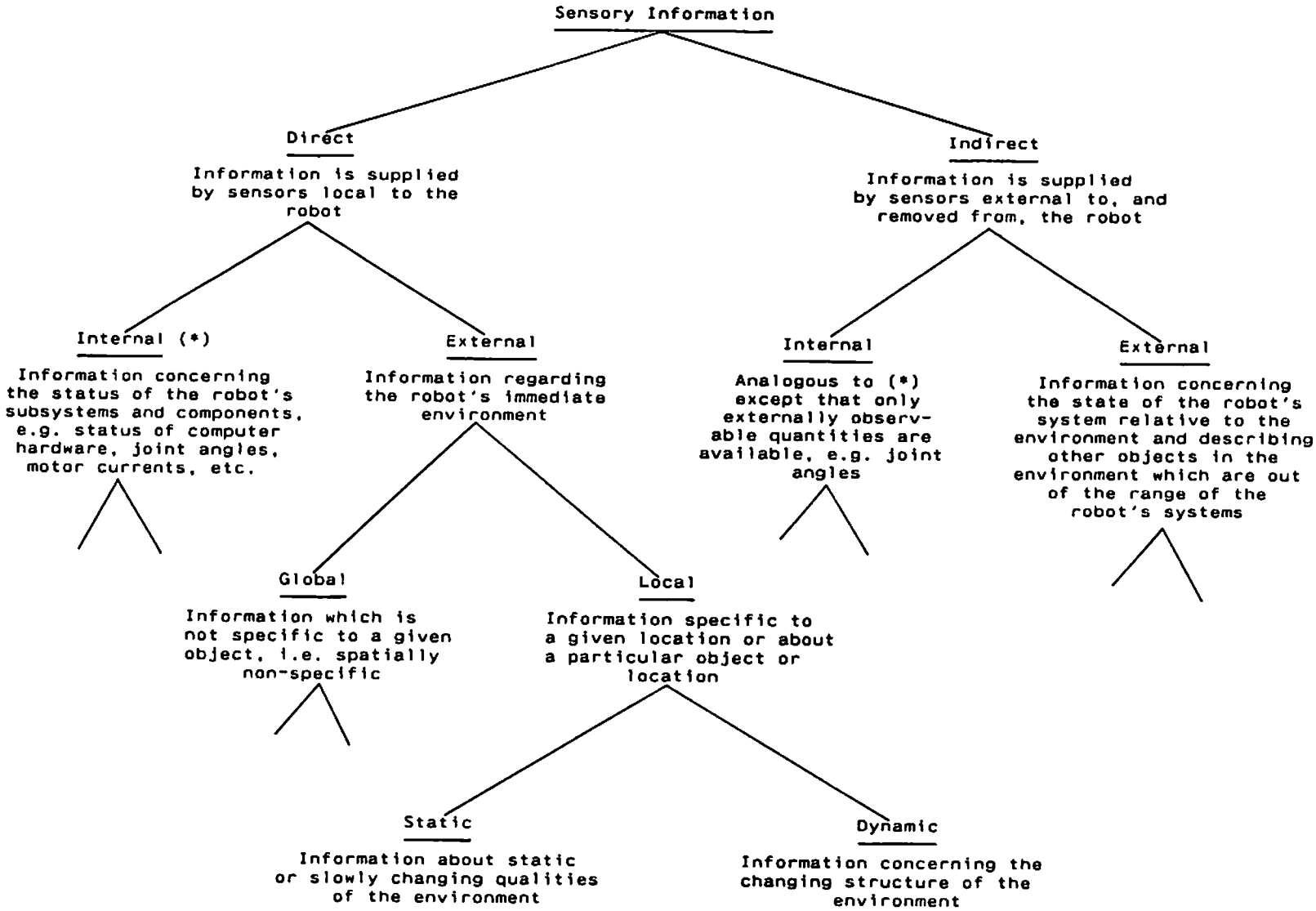
1.2.2 Human Touch Physiology

To this point, the discussion has focused on abstract classifications of sensory information. It is now appropriate to consider more concrete issues surrounding sensing. The tactile system of humans provides an excellent study case for these final classes. The following material includes discussions of both the structure and function of the receptive system of the skin. While only the response information relates to the present discussion, the outline of the structure is relevant to the succeeding chapters. For more detailed discussions of tactile sensory systems, the reader is refered to [81] and [71].

Touch includes all of the sensations existing at the interface between the human body and the environment. The sense of touch in humans encompasses a great number of submodalities including temperature, pain, and nociception as well as the area of shape or pressure discrimination. Since the sensors available to robots most closely functionally emulate the latter, the discussion here will concentrate only on the mechanism and characteristics of this submodality. Touch is localized in the thin layer of cells which marks the boundary between the body on the inside and a harsh environment on the outside. This layer, the

Figure 1.1

A Taxonomy for Sensory Information. The first distinction discriminates between data supplied by sensory systems local to the robot, *DIRECT*, from that supplied by remote systems, *INDIRECT*. *INTERNAL* information concerns parameters of the device itself while *EXTERNAL* relates to the environment. The *GLOBAL/LOCAL* distinction differentiates information specific to a location or object, *LOCAL*, from that which pertains to general characteristics of the environment, *GLOBAL*. The final classification of *STATIC* versus *DYNAMIC* relates the rate of change of the characteristic of interest.


Sensory Information
Direct
Information is supplied by sensors local to the robot
Indirect
Information is supplied by sensors external to, and removed from, the robot
Internal (*)
Information concerning the status of the robot's subsystems and components, e.g. status of computer hardware, joint angles, motor currents, etc.
External
Information regarding the robot's immediate environment
Internal
Analogous to (*) except that only externally observable quantities are available, e.g. joint angles
External
Information concerning the state of the robot's system relative to the environment and describing other objects in the environment which are out of the range of the robot's systems
Global
Information which is not specific to a given object, i.e. spatially non-specific
Local
Information specific to a given location or about a particular object or location
Static
Information about static or slowly changing qualities of the environment
Dynamic
Information concerning the changing structure of the environment

skin, varies in thickness from 0.5mm over the eardrum to more than 5mm on the hands and feet. The tactile sense is anything but uniform in its covering of the body surface. Differences exist between the hairy and non hairy, or glabrous, skin areas with most of the receptors in the glabrous skin. Even within the glabrous skin areas of the hands and feet, the receptor densities and perceptual acuity vary from one area to another. The discussion that follows concentrates on the characteristics of the glabrous skin and the digital skin areas in particular.

For purposes of this discussion, the tactile system consists of a set of receptors embedded in an elastic containing medium, the dermis, covered by a relatively tough protective layer, the epidermis. This two-layer structure is supported by a flexible underlying layer, the hypodermis or superficial fascia, which provides both a compliant support and an interface to the muscles. All of this tissue is connected to the skeleton which provides a rigid base structure for the system. The receptors are connected to the spinal cord and the central nervous system by bundles of afferent nerve fibers. The properties of the skin -- that is, the combined properties of the epidermis and dermis -- are of interest in relation to the mechanical properties of the sensor pad used in this work. Knowledge of the structure and function of the various receptors is of interest in comparison with the characteristics of the transducer used here. Finally, the response characteristics of the afferent nerve fibers connecting the receptors to the CNS are of interest relative to the processing of tactile images from robot sensors. Each of these three areas are discussed below.

1.2.2.1 Structure of the Skin

The skin consists of two major divisions, the dermis and epidermis. The superficial layer of the epidermis provides the actual interface between the human and the environment. This layer, known as the stratum disjunctum or epitrichium, is composed of dead cells and is characterized by irregular and broken connections with the underlying layer. The cells in this layer slough off and become part of the environment. The next proximal layer is termed the stratum corneum and shows more structure than the stratum disjunctum but is still composed of dead cells. The stratum lucidium is the next deepest layer and is seen as a thin layer found only in the palms of the hands and soles of the feet. At this point the cells are small and tightly packed. The stratum granulosum, or layer of Langerhaus is next and marks the beginning of cell death and the outermost location of receptors. The living cells in the epidermis are in layers termed the stratum malpigii, Malpigian layer, or rete mucosum. The bottom layer of the epidermis is a single layer of cells termed the stratum basal or stratum germinativum. Epidermial cells are born in the stratum malpigii and migrate outward to be lost finally from the stratum disjunctum over a period of between 20 and 30 days.

The epidermis is separated from the dermis by a thin non-cellular layer known as the basement membrane or basal or adepidermal lamina. This layer serves the function of connecting the two skin layers together and is composed of a network of extracellular fibrills plus glyco-proteins [174]. In the digital skin,

the surface of the epidermis is covered by the fine ridges which produce finger prints: see Appendix B for a further discussion on fingerprints. These ridges are actually foldings in the epidermis and produce similar ridges and valleys in the underlying surface. The downward pointing, i.e. proximal pointing, ridges are known as rete pegs and mesh with ridges in the distal surface of the dermis called primary dermal papillae. It is felt that these interlocking ridges aid in resisting shear forces between the two skin layers. Other mechanisms include the attachment of the basement membrane to the two skin layers and sweat ducts which run from glands deep in the dermis to the skin's surface.

The dermis has essentially two layers: the papillary layer distally and the reticular layer proximally. The papillary layer consists, as one would expect, of the dermal papillae and contains vessels, nerves, sweat glands, hair follicles, and certain receptors. The reticular layer is essentially all of the material between the papillary layer distally and the hypodermis proximally and contains some fat cells. As indicated above, the hypodermis connects the dermis/epidermis structure to the muscles and bones.

Petit and Galifret [164] have studied the mechanical characteristics of glabrous skin in the human fingers and shaved skin of the rat. The skin appears to be a stable platform for the receptors. Upon closer look, it is seen that the surface of the skin undulates with the respiration of the individual with an amplitude of between 10 and 30 mumeters. This implies that the actual location of a given receptor relative to the underlying bone structure is most likely not available to the central nervous system (CNS). Furthermore, the movement of the skin results in stresses in the skin which should produce output from some of the receptors. Petit and Galifret also found that the force required to maintain a given indentation of the skin decrease as the probe is held against the skin. That is, the skin tends to relax with time. This effect was measured for up to eight seconds with the most pronounced effects seen during the first two. Once removed, the indentation from the probe remained. In some cases, the indentation remained for three to four minutes.

From this work we see that, mechanically, the skin actually provides a constantly moving platform for the receptors, that it does not act as a true spring but rather that it relaxes with continued application of a force or deflection, and that it exhibits a great deal of hysteresis in that an indentation remains after removal of the stimulus. Despite these problems, humans are capable of fine tactile discrimination, and that in very small time frames. This leads one to believe that useful information can be obtained from a system with inherent problems of hysteresis and time-varying response characteristics. This view needs to be borne in mind during the discussions of the analogous parameters of the sensor used here.

1.2.2.2 The Receptors

There are five basic receptors in the glabrous skin of humans: the "free" intraepidermal nerve endings, the Merkel discs, papillary nerve endings, Meissner corpuscles, and the Pacinian corpuscles. The previous list is ordered in increasing

distance from the environment. The reader is referred to an excellent article by Quilliam for an overview of the physiology of tactile receptors [174]. The following discussion will outline the structure and distribution of each of the receptors. A clear picture of the functional characteristics of all but the Pacinian corpuscle is not yet available but the known relationships are discussed in the next section.

The "free" nerve endings and the Merkel discs are actually both intraepidermal endings since they reside in the epidermal layers. The free endings can be traced to nerves from the superficial dermal nerve plexus. These nerves appear as myelinated fibers in the primary dermal papillae; they continue along the basement membrane as unbranched, sheathed fibers. Upon entering the epidermis, they lose their Schwann cell sheath and myelin and continue unbranched into the stratum granulosum. Since there is no branching of the nerve fibers, it appears that the information transduced by the free nerve endings may be directly available to the CNS.

The Merkel discs are innervated by fibers of the same origin as the free nerve endings. The discs, however, reside in the stratum germativum, much deeper than the free nerve endings. Each disc may be innervated by many nerve fibers and each nerve fiber branches and may service several discs. Thus in the case of the Merkel discs, there is a many-to-one and a one-to-many receptor to nerve mapping. That is, a particular nerve fiber receives input from a number of discs and a given disc provides input to a number of nerves. It has been postulated that the free nerve endings are actually nerve fibers which originally innervated Merkel discs. As a disc migrates toward the periphery and dies, some of the nerve fibers may survive to become free endings. The remaining three receptors are located in the dermal layers.

The papillary nerve endings are the simplest of the remaining three due to the fact that they exhibit no specialized end structure. As with the first two receptors, the papillary nerve endings are derived from nerves in the superficial dermal nerve plexus. The sheath is lost and the fibers end in very thin ultraterminals, loops, spirals, balls (hederiforms), or lattice structures. The final two receptors are notable due to their specialized end organs and, especially in the case of the Pacinian corpuscle, have been heavily studied.

The Meissner corpuscle appears only in Man and higher apes and only in the fingers and toes even in these species. They appear as ovoid bodies typically 80 micron in length and 30 micron in diameter. The usual location is attached to the top of a papillary ridge with its major axis oriented perpendicular to the skin's surface.

Each capsule has a laminated construction with heavy innervation of a highly structured core. The myelinated nerve fibers begin in the superficial dermal plexus and branch to innervate several corpuscles. Each corpuscle is typically serviced by two to three fibers although in some cases as many as six to seven fibers have been found in a single corpuscle. These fibers form a fasciculus of supply which enters the capsule at the proximal end. The myelin sheath is lost as the fibers move through the capsule. In addition to the Meissner corpuscles, these nerve fibers may also have other endings. Thus the many-to-one

and one-to-many mapping relationship seen for the Merkel discs applies in this case as well. The situation is confounded, however, by the fact that the nerve supply to the Meissner corpuscles also receives information from other sensory endings.

The core consists of several lobules of cells located one on top of the other with the lobule organization increasing with decreasing distance to the periphery. The most distal lobule is composed of a set of discoid cells stacked on top of one another similar to a pile of poker chips. The bottom, or most proximal, lobule contains irregularly shaped cells. Myelinated fibers enter each level, or lobule. Here the sheath is lost and the fibers branch and wind around the cells of that lobule. Some of the branches may exit and innervate the next most distal lobule. One could posit that the most tuned section of the corpuscle is the most distal lobule which not only contains the most highly structured cells but is also the closest to the environment.

The final receptor is known as the Pacinian corpuscle. These exist as ovoid structures typically 1.5mm in length and 0.5mm in diameter. They are the largest single receptor found in the human body. A typical human has roughly 2000 corpuscles with about 1/3 located in the fingers and toes. The remaining 2/3 are sparsely distributed throughout the body with some even appearing in deep tissues, e.g. the pancreas. The corpuscle has a thin capsule which surrounds a wide outer core described as having an "onion skin" structure of thin skins separated by pockets of fluid. This outer core contains a compact inner core composed of two parallel stacks of cells. The cells are shaped in such a manner that there is a channel down the center of the core between the two stacks of cells. This channel is the site of innervation.

Each Pacinian corpuscle is innervated by exactly one myelinated fiber and that fiber proceeds unbranched from the spinal cord. Thus each corpuscle has direct input to the spinal cord and brain. The fiber enters at one pole of the capsule and loses its sheath as it enters the inner core. At this point there is a small constriction and then the bare fiber travels straight along the channel between the two stacks of cells. At the distal end, the fiber may branch to form ultraterminals but does not leave the inner core. Probably due to their size, Pacinian corpuscles have been the target of many studies. They respond to vibratory stimuli and each appears to be tuned to one of two frequencies. Let us now direct our discussion to the response characteristics of the mechano-receptive system. The reader is refered to Appendix B which contains both a more complete discussion of the Pacinian corpuscle and some ideas regarding how an electro-mechanical device which emulates this corpuscle might be constructed.

1.2.2.3 Response Characteristics

It it not possible to associate a given conscious perception with a particular receptor. With the exception of the Pacinian corpuscle it is not even possible to associate a particular response to a given receptor. One of the problems is the fact that the skin, considered from a mechanical point of view, affects the

response of the receptors in the form of filtering, attenuating, etc. In addition, the fact that the skin is a physical material implies that a given deflection will result in the deflection of the skin around the stimulus and thus will elicit activity from receptors not directly under the stimulus. In later chapters, this phenomenon is referred to as the spread function of the material during discussions of the response characteristics of the robot sensor used in this work. The many-to-one and one-to-many mappings found in two of the receptors as well as the fact that more than one type of receptor may provide input to a given fiber also confounds the problem of identifying the exact response characteristics of a given receptor. What is more important, however, is not the response of a particular receptor class but rather the information provided to the central nervous system, the CNS.

Studies of the response characteristics of afferent nerve fibers to mechanical stimuli has led to a simple classification of the nerves. Figure 1.2 contains the classification scheme with receptive field size and adaptation rate as the discriminating factors. This leads to four classes: RA and PC for fast adapting

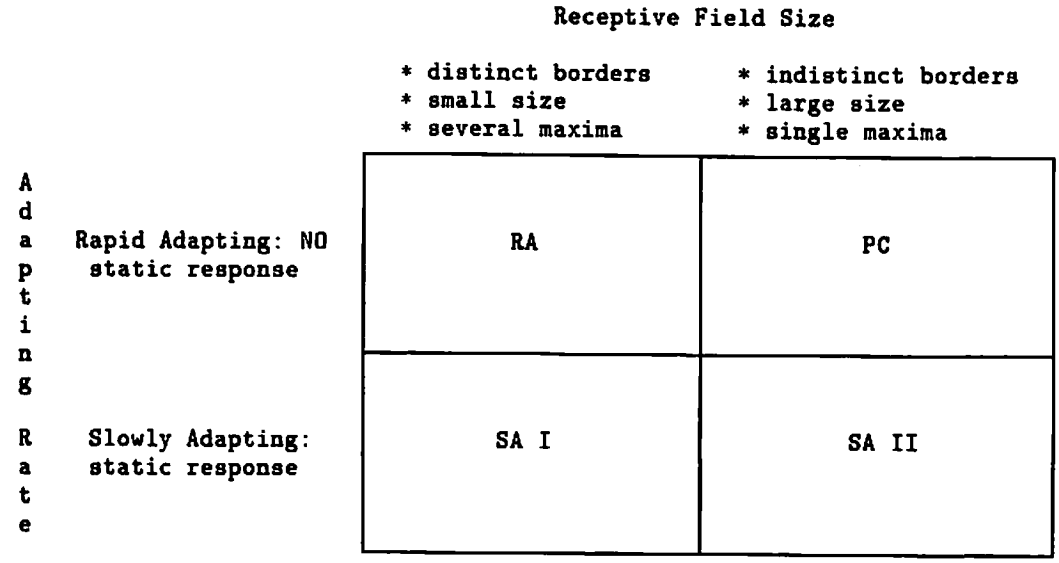


Figure 1.2
Nerve classification based on receptive field characteristics and adaptation rate. The horizontal axis discriminates between two receptive field size classes while the vertical axis discriminates the adaptation rate. These characteristics are used to develop the lower two categories in the Sensory Information Classification Scheme in Figure 1.1. The Meissner corpuscle is thought to be associated with the RA fibers and the Pacinian corpuscles with the PC units. Other receptor/response matchings are unknown.

and SAI and SAII for slowly adapting. The RA nerve fibers have a median receptive field diameter of about 4mm which corresponds to an area of 12.6 square mm. The median figure is slightly misleading since the median sizes of the RA receptive fields vary by a factor of 1.7 between the finger and the palm. In the palm the distribution is uniform with fields up to 19mm in diameter while in the fingertip the field sizes are concentrated around 2-3mm. Results from two-point acuity tests [206] suggest that the receptor density for the four fiber types should increase toward the periphery. The RA receptor density varies in two rather distinct steps from 25 units per square cm in the palm to 141 units per square cm in the fingertip. The Meissner corpuscles are thought to be associated with these fibers and the multiple sensitive maxima characteristic of the RA nerves would then correspond to individual corpuscles. The PC units have extremely large receptive fields which may cover an entire finger and are thought to be associated with the Pacinian corpuscles. These units, as in the cases of the SAI and SAII units, show no significant change in receptive field size from the palm to the fingertip. However, the receptive field density varies from 9 units per square cm in the palm to 21 units per square cm in the fingertip. Recall that each Pacinian corpuscle was innervated by exactly one fiber and that fiber proceeded unbranched to the CNS. This could account for the single sensitive peak in the receptive field of the PC unit. The SAI receptive field characteristics are very similar to those of the RA units and the Merkel discs are thought to be the associated receptor. The receptor density varies from 8 units per square cm in the palm to 70 units per square cm in the fingertip. SAI nerves exhibit irregular firing for constant stimuli. The remaining units belong to the SAII class and show continuous discharge for static stimuli but have a much larger receptive field than the SAI units. The SAII units continue to respond after removal of the stimulus, a form of hysteresis. The trend in the receptor densities for the SAII units is the reverse of the previous classes with typically 16 units per square cm in the palm and only 9 units per square cm in the fingertip. No receptor association for humans has been suggested but in cats the Ruffini endings are associated with the SAII fibers.

Quilliam [174] suggests that touch has two phases. The first phase is one in which initial contact is made with an object and the situation is classified according to dangerous/safe criteria. During the second phase the fingers and hand are used to manipulate the object to determine features of its surface. It is easy to see that the rapidly adapting, large receptive field units, PC, may serve the first phase since they indicate that something has occurred somewhere. The classification during this phase requires information from other submodalities but the PC unit response is enough to indicate presence of a stimulus. The second phase requires a great deal of information and may be served by the remaining classes of mechanoreceptors, see below. Vallbo [206] feels that tactile sensibility has only one function: to locate skin indentations and/or the parameters of those indentations. This viewpoint may hold true for the mechanoreceptors but certainly does not include the other submodalities of touch. At this point it is useful to analyze two simple stimulus situations in regard to

the responses they will elicit from the four classes of nerve fibers.

The first situation to consider consists of a single point indentation near the tip of the index finger. The stimulus is stationary in both time and indentation magnitude. Since the stimulus is stationary, the rapidly adapting units will fire only as the stimulus is applied. In particular, PC units located all over the finger (and possibly the palm) will fire indicating that something is happening. Only RA units in the area of the stimulus will fire due to the smaller receptive field size. Note that this firing will occur only as the stimulus is applied and released, not while it is stationary. The SAI and SAII units will fire continuously as long as the stimulus is present. The SAI units under the stimulus will fire due to their small receptive fields while, analogous to the PC units, the SAII nerves from much of the finger (and perhaps the palm) will fire. Thus, information regarding the onset and conclusion of a stimulus event is supplied by the rapidly adapting units with the RA units providing much more location-specific information. The slowly adapting unit provide positional information, again with the "grain" determined by the receptive field size.

A more interesting situation occurs when this same stimulus point begins to move. In this case the PC units, again from all over the finger and palm, fire because of their large receptive fields. They will fire continuously since the stimulus is constantly changing. The RA units in the immediate vicinity of the stimulus will fire as the stimulus moves. These units fire as the stimulus approaches and departs thus providing moving edge information. The SAI units will fire only when the stimulus is essentially directly on top and the SAII units will fire analogously to the SAI units except with a larger field. Thus the PC units could be described as providing global stimulus movement information, the RA units as providing localized movement information (e.g. edge detection), the SAI units as localized positional information, and the SAII units as global positional information.

1.2.3 Taxonomy: Part 2

The previous discussion of touch physiology leads directly to the final two levels of classification in the hierarchy. These are given by the two axes in Figure 1.2. However, instead of interpreting the characteristics as being of a particular sensor or receptor as in the figure, the classifications are applied to the type of information supplied to the system.

The first level provides a distinction between *GLOBAL* and *LOCAL* information. We have seen that there are two classes of nerve responses in the tactile system. One is characterized by response fields which are very large and cover a great portion of the sensory surface. These units respond to the presence of a stimulus regardless of its placement within the receptive field. They provide immediate information relating the fact that something is happening at the periphery. The exact location, and in some cases magnitude, of the stimulus event is unavailable through these units. Translated into more general terms, *GLOBAL* sensory information refers to information which concerns specific non-spatially characteristics of the environment. The temperature of the

atmosphere and the overall brightness of the environment are two examples of *GLOBAL* information.

The second class of nerve responses is typified by elements whose receptive fields are very limited in spatial extent. Here the information is *LOCAL* in nature. In the tactile case, these units exhibit small receptive fields and these fields are statically located and cover correspondingly small areas of the sensory surface. The activity of these units may be interpreted as indicating the exact position of a stimulus on the sensory surface. Again in more general terms, *LOCAL* information relates the properties or characteristics of a specific element or location in the environment. The temperature of an object and the color of a particular region in the visual field are examples of *LOCAL* information.

The vertical axis of the table in Figure 1.2 categorizes the units based upon the temporal properties of their activity. One class of afferents discharges continuously while the application of the stimulus is static while the other class fires only when the stimulus changes. This leads to the classification parameters of *STATIC* versus *DYNAMIC*. As in the discussion of the *GLOBAL/LOCAL* classification, the emphasis here lies on the characteristics of the stimuli as opposed to fiber responses.

STATIC sensory information includes information about invariant characteristics of the object or environment. This includes information about both characteristics which are static, e.g. the mass of an object, and unchanging relationships between the stimulus and the sensing system, e.g. point of contact by unmoving objects. In the tactile system, the SA units respond in a *STATIC* fashion providing a continuous stream of information regarding the static interaction between the sensory surface and the object. The static tactile image processing discussed in Chapter 4 is designed to yield *STATIC* (and, in fact, *LOCAL*) information from tactile images.

The other companion category at this level of the taxonomy includes *DYNAMIC* information. The emphasis here is on information regarding the changes between the sensory system and the stimulus. Recall that the RA and PC units in the human tactile system fire only while a stimulus in changing. This information includes both changes in the object, e.g. swelling or deformation, and changes in the relationship between the object and the sensing system, e.g. movement. One facet of the dynamic tactile image processing discussed in Chapter 5 deals with this form of information.

A classification scheme has been proposed which begins by differentiating between information supplied by the robot's sensory system and that supplied by external sensors (*DIRECT* vs. *INDIRECT*.) The next level of classification distinguishes between information regarding characteristics within the robot system from those of the environment (*INTERNAL* vs. *EXTERNAL*.) The classification was refined one more step to differentiate information relating *GLOBAL* events and characteristics from those which are purely *LOCAL*. The final level distinguishes information regarding the non-changing parameters of the environment or robot/object relationship (*STATIC*) from information pertaining to the changing nature of the world (*DYNAMIC*.)

This classification scheme, as with any scheme, achieves its usefulness from the fact that it allows one to organize thoughts regarding sensory information and provide some access, when viewed in light of a sensor taxonomy, to a mapping between a desired piece of information and a sensor or sensor technology which may provide that information. The next step then consists of developing a taxonomy for robot sensors.

1.3 Taxonomy of Robot Sensors

In the robot domain, a sensor is a device which transduces a physical quantity into a form amenable to utilization by a machine. Several authors have presented discussions regarding the sensors available to machines, [181], [179], [146], [149], and have proposed various classification schemes. A reasonable approach to a discussion of sensors is to delineate the various classes of transducers. The class structure presented here is based upon the quantities transduced by the sensors and the manners in which the sensors interact with the environment. It is similar to that in [146], but inverted and has special emphasis placed on the tactile domain.

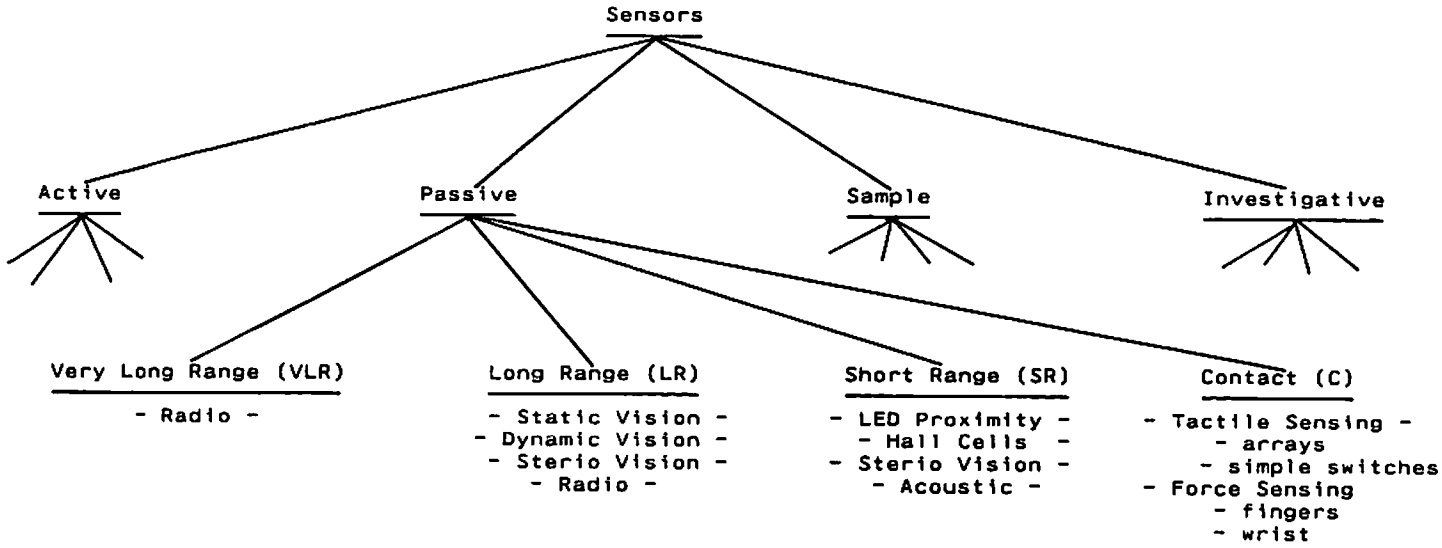
Robot senses provide information regarding internal quantities such as joint angles and motor currents as well as external quantities, e.g. through vision, touch, and rangefinding. However, with the one notable exception of wrist force sensing research (see below), the term "robot sensors" is typically thought to include only those devices which are local to the robot, i.e. those which provide *DIRECT* information, and devices which transduce external events, i.e. those which yield *EXTERNAL* information. Note that a number of different sensors may be used to determine a particular piece of information: for instance, vision can be used to deduce joint position. The following discussion focuses on these devices.

Sensors are categorized depending upon a number of factors, see Figure 1.3. The highest level of the classification discussed below involves the manner in which the sensor transduces the target quantity and includes: *ACTIVE*, *PASSIVE*, *SAMPLE*, and *INVESTIGATIVE*. In the next level, sensors are categorized depending upon how "intimate" the sensing device must be with the object being sensed or physical quantity being transduced.

A sensor is termed *PASSIVE* if it relies only on energy from the object or environment in order to transduce the quantity of interest. In this case, the energy must be reflected, transmitted, or emitted by the object. This is in contrast to an *ACTIVE* sensor which emits energy into the environment and compares the returned signal with the emitted signal to obtain an indication of the physical quantity of interest. Both *PASSIVE* and, in the ideal case, *ACTIVE* sensors leave all portions of the physical environment undisturbed. A *SAMPLE* sensor varies from the two above in that it physically captures a small portion of the environment and analyzes it for the feature(s) of interest. An olfactory sensor would analyze a sample of the atmosphere to provide a description of the "smell". A final class may be termed *INVESTIGATIVE* and encompasses the case in which the transducer must be extended into the environment in the form of a probe. This is meant to include cases such as a tactile sensor used

Figure 1.3

A taxonomy for robot senses. The highest level of classification relates the way in which the sensor interacts with the environment. A *PASSIVE* device relies on energy provided by the environment while an *ACTIVE* sensor injects energy into the environment and transduces the reflected signal. A *SAMPLE* sensor captures and analyzes a portion of the environment and, finally, an *INVESTIGATIVE* is one which is used to probe the environment. The second level of classification involves the distance at which the sensor functions. The spectrum runs from *CONTACT* sensors to very long-range sensors, *VLR*, with the intermediate classes of proximity sensors, *SR*, and intermediate range devices, *LR*.


Sensors
Active
Passive
Sample
Investigative
Very Long Range (VLR)
- Radio -
Long Range (LR)
- Static Vision -
- Dynamic Vision -
- Sterio Vision -
- Radio -
Short Range (SR)
- LED Proximity -
- Hall Cells -
- Sterio Vision -
- Acoustic -
Contact (C)
- Tactile Sensing -
- arrays
- simple switches
- Force Sensing
- fingers
- wrist

to explore the immediate environment and not instances where, for example, a tube is extended to collect an uncontaminated sample of the atmosphere. The ordering given above of *PASSIVE*, *ACTIVE*, *SAMPLE*, *INVESTIGATIVE* is in increasing interaction with the environment. In the case of a *PASSIVE* sensor, the transducer essentially only monitors the environment while an *INVESTIGATIVE* sensor interacts with the environment in a dynamic fashion. It is important to note that is some cases a particular sensor may fall into one category when used in one fashion and into another category when used differently. For example, a tactile array sensor -- when used to cover the work space -- is a *PASSIVE* device in that the stimulus object's energy and movement cause interaction with the device. When the sensor is attached to the fingertip of a robot's gripper and used to probe the environment it falls into the *INVESTIGATIVE* category.

The second distinguishing characteristic relates the proximity of the sensor to the stimulus being measured. At one end of the spectrum are those devices which must be in direct contact with the object exhibiting the characteristic to be transduced. These devices are, obviously, termed *CONTACT*, or *C*, devices. The other extreme includes the set of devices which transduce the phenomenon of interest from a great distance and are termed *VERY LONG RANGE*, or *VLR*, devices. Between these two classes are two rather arbitrarily defined classes. The class closest to the *CONTACT* end of the spectrum contains sensors which work in close proximity to, but not in contact with, the object being sensed. This class is termed *SHORT RANGE*, or *SR*. (The term "object" is used throughout to indicate the possessor of the characteristic to be measured. It is easier to discuss and visualize these classifications when one uses a concrete, bounded object as the target of the sensing rather than an immaterial quantity such as visible light.) The final class, *LONG RANGE*, includes those devices which operate at distances greater than those of the SR sensors, i.e. greater than a few meters, and at distances shorter than the *VLR* devices, i.e. within a few kilometers.

Various sensors are outlined below with the discussion generally following the structure of the taxonomy as presented in Figure 1.3. Within these major classes, sensors in each of the subclasses are discussed.

The most prominent sense to the majority of humans and the area receiving the greatest attention in the past, is that of vision. Here vision is defined as the acquisition and processing of two-dimensional (possibly time-varying) arrays of numbers representing the light intensities of a scene in the real world. The basic paradigm involves taking a "snap shot" of the scene currently in the view of the camera. This image is then processed to yield a description of the objects in the scene and their relationships to one another. Under this paradigm, the camera is termed *PASSIVE* since it utilizes electromagnetic radiation solely from the environment. In addition, it may belong to the *VLR*, *LR*, or *SR* classes depending upon the manner in which it is used. During the past several years a great deal of effort has been directed toward the problems involved with static scene analysis as evidenced by the work of [36], [141], [89], [90], [91], and [223] at the University of Massachusetts, [102], [103], and [132], [130] at the Massachusetts Institute of Technology, [182] at Maryland, [198] and [20] at SRI

International, [24] at Stanford, [170] and [111] at Carnegie Mellon University, and a large number of other researchers -- a good sampling of this work is provided in Computer Vision Systems, edited by Hanson and Riseman [90] see also [17] and [56].

Machine vision research has focused on several specific application domains: remote sensing data (typically from satellites), outdoor scenes (including military applications), biomedical image processing, and limited industrial settings. The applications of machine vision research and techniques to industrial environments are, however, continually increasing in both number and sophistication. In most robotic applications the image of the world is thresholded to produce a binary image in order to reduce the storage required for the image and the amount of computation required during the succeeding processing stages. Due to the relevance of vision to the subject of this work, only a brief discussion of current applied vision research efforts is appropriate. For further discussions of vision research applied to the robot domain, the reader is referred to [172], [6], [7], [5], [15], [80], [31], [207], [163], [162], [220], [155], [96], [201], [27], [179], [140].

While the majority of work has focused on military domains and outdoor scenes, there is a growing numbers of industrial applications where some degree of vision is employed. Most of the work applied to robotics has been limited to the visible spectrum however research is in progress which utilizes alternate spectra, e.g. infrared, x-ray, etc. The present use of vision by robots assumes the form of: take a picture, analyze, plan, and then blindly act with no regard for changes in the environment. An early example of such a scheme was Shakey the robot at SRI. The current trend in most industrial robot vision systems consists of calculating statistics for any dark objects found on a light background (or vice versa) [4]. The Automatix system is an exception in that it is designed to handle both multiple thresholds and some grey-level processing [178]. From these feature vectors, the likely identity of the part can be chosen from a group of pre-specified parts. In the completely static case, the orientation is determined from the feature values and the model of the part thus allowing a robot arm to obtain the part.

In the case of systems such as CONSIGHT [212], [101], [100], (see also [167] for a discussion of a similar system) a single binary image is constructed by a linear camera as the object passes by on a conveyer. A light stripe projector is employed by the camera system to illuminate the scene. Since the system emits energy into the environment, this configuration is termed *ACTIVE*. The domain of this system is the industrial work place and the nature of this domain coupled with the effective range of the light projector makes this system fall into the *LR* class. The image is processed to locate the position, as defined by a particular feature of the object, and the orientation. The speed of the conveyer is monitored and the location of a part can be predicted from the speed and processed image. Thus a manipulator can track and intercept an object.

All of the work previously cited involves the processing of static images. Most of the work applied to robotics has been limited to the visible spectrum;

however, research is in progress which utilizes alternate spectra, e.g. infrared, x-ray, etc. Motion vision, i.e. sequences of static images, is beginning to receive attention [215], [216], [120], [121], [122], [168], but has yet to be utilized in the robotics domain to any extent. Thus static vision, as typified by the utilization of binary or grey scale images from optical cameras, is usually *PASSIVE* and provides a 2D image of the 3D world as variations in the sensed intensity of energies in a particular portion of the electromagnetic spectrum. We have seen, however, that machine vision can be *ACTIVE*. Vision is typically thought of as applying to objects and scenes on a size scale comparable with scenes encountered in everyday life. However, machine vision also encompasses the world as seen through a microscope or from a satellite. Thus, depending upon the situation, vision may be termed *VLR*, *LR*, or *SR*.

Vision as described above lacks explicit information regarding the third dimension of the environment. The images are processed to extract information about the scenes. Rangefinding sensors are utilized to provide this information directly [110]. In this case, the information from the sensor indicates the distance from the sensor to points in the world. Acoustic or laser techniques are typically used to construct depth images of the environment. In these cases, energy is emitted into the environment and its reflection sensed. Thus these devices are *ACTIVE*. As in the case of the CONSIGHT system, the effective range of these systems is limited, thus making them fall into the LR and SR classes. Alternatively, stereo vision can be used to derive the depths via triangulation. Two static images of the same scene are collected from slightly different positions [131], [110], [219]. The location in each image corresponding to a given feature and the known parameters of the optical system are used to calculate the distance of the feature. These systems are *PASSIVE* and tend to be either *VLR* or *LR*. One of the basic problems in this paradigm (as well as dynamic image processing) is known as the stimulus-matching problem, i.e. the task of finding the same feature in both images. Once a point has been located in both images, its depth can be determined. Moravec [137], [138] used a similar approach in which nine images were taken from known positions along a track. The viewing axes were parallel in his scheme where in true stereopsis the axes intersect at some point in space. Dynamic image processing techniques, see [215], [168], can be used to extract this type of information as well. The technique known as optic flow [77], [78] [123], [121], [169] also has great potential in this respect. Again, the techniques just mentioned are *PASSIVE*. It should be noted that the previous discussion assumed that the camera was fixed relative to the robot. An interesting alternate paradigm involves moving the camera in a specified manner and using knowledge about the motion and the resulting image data to extract information about the robot's spatial relationship with the environment, i.e. placing a camera in the robot gripper. In such a case, the sense may be termed *INVESTIGATIVE*.

There are several rangefinding techniques which are *ACTIVE*. Laser and acoustic ranging both fall into this category [110]. In the case of laser imaging, a laser beam is used to scan the environment. The reflected light is sensed by the system and the time of flight is used to determine the distance from the

sensor to the reflecting object. An alternate approach involves triangulation wherein the laser beam is used to illuminate a point in the environment. The geometric parameters of the laser and the optical sensor are used to calculate the three dimensional coordinates of the point. Examples of rangefinding systems and techniques are provided in [166], [148], [150], [190], [109], and [188], see also the review in [110]. These systems can produce a 2D image of the world where a value in the image is indicative of the distance to the nearest object in the environment along a line from the point in the sensor plane. Acoustic imaging functions in a similar manner. In the simplest case, a set of pulses at varying frequencies is emitted into the atmosphere. The sensor monitors the echo and measures the time of flight for the peak of the echo. This provides the distance to the largest near object. Radar and Sonar systems are extensions of this simple paradigm in which the direction of the returning signal is monitored to provide a distance map similar to the laser technique. In the cases mentioned above, the sensor emits energy into the environment and monitors the reflected energy to determine distance.

There have been efforts directed toward marrying the vision work with the *ACTIVE* approach of the rangefinding techniques. As discussed above, typical vision systems rely on ambient lighting. Here the illumination is controlled by the sensing system. Structured light is projected into the environment, a picture taken, and the warped pattern of light is compared to the projected pattern. The form of the warping is indicative of the spatial characteristics of the objects in the environment. Albus, among others, uses this approach [208] and the CONSIGHT system [212], [100], [101] can be viewed as functioning in this paradigm.

The vision and rangefinding techniques are used to provide information regarding the machine's immediate surroundings. For example, Bolles and Fischler used range data to locate cylinders in images of jumbled industrial parts [30]. Both can be used to provide detailed information about a localized area (foveation) as well as less information about a much wider view. There are some exceptions where vision is being used to examine microscopic parts. The interest in robotics is not necessarily in measuring the distance to the part but in understanding the identity and orientation of the parts (and features of parts) whose sizes are of the same order as the size of the robot.

Non-contact sensing techniques have also been developed for use in determining short distances [152], [108]. This sensing is known as proximity sensing and can be either active or passive. The reader is refered to [115] for an in-depth review of this area. One approach to *ACTIVE* proximity sensor consists of an LED/phototransistor pair [40], [41], [21], [22], [62]. This can be extended to an array of detectors as in the work of Okada [151]. Typically, an LED emits light into the environment and the phototransistor receives light reflected from objects. Crosnier used this approach for sensors inside the fingers of a gripper to detect movement of the surface due to contact with an object [48]. In another approach, a jet of air is blown into the environment and disturbance of the flow is monitored [88]. In all cases, the sensor provides distance information for an

object near to the sensor. The use of an electromagnetic field to transduce eddy currents in nearby conductive plates is an example of a *PASSIVE* proximity sensor. All of these proximity sensors work on a distance scale which is much smaller than that generally used in vision and provide information only when an object is very near to the transducer.

The sensors just described all fall into the general class of *NON-CONTACT* sensors which includes the classes of *VLR*, *LR* and *SR* sensors. The remainder of the sensors to be considered are of the *CONTACT* type.

The most immediate aspect of the world surrounding the robot is the atmosphere. Transducers of the temperature, density, moisture content, and composition of the atmosphere are all contact sensors. A temperature sensor can be *PASSIVE* in nature, i.e. a thermometer, or *ACTIVE*, i.e. a thermal conductivity sensor. The latter provides information useful in the analysis of the composition of objects with which the robot is in contact. Sensors transducing the density, moisture content, and composition of a gas may be classified as *SAMPLE* sensors since a small portion of the atmosphere is captured and analyzed.

Aside from the atmosphere, contact sensors are used to provide information regarding objects in the immediate vicinity of the machine. Proximity sensors have already been mentioned in the *PASSIVE* and *ACTIVE* cases. A whisker sensor is a proximity transducer of the *INVESTIGATIVE* type. Here a fine, flexible shaft acts as a probe. When the probe contacts an object, the shaft bends and this information is available to the robot [66].

Once in contact with an object, tactile sensation comes into play. There are three general types of information of interest: the physical parameters regarding the composition of the object, the parameters describing the interaction with the object, e.g. the forces and torques, and the object's shape. The former class includes such quantities as thermal conductivity, hardness, moisture content (useful in understanding slip), mechanical resonance properties, eddy current transduction, mass, location of centroid. Of these properties, only the latter two are unavailable from a purely local sensor. The mass and centroid can be derived from internal sensors.

The desired information regarding the interaction between the robot and an object can vary in complexity from simple contact detection to force and torque parameters. In the simplest case, interaction between the robot and its environment takes the form of the robot bumping into an object. A contact switch located on the surface of the robot can be used to determine collision with an object contacting the robot at that point. Knowledge of contact may be sufficient in many instances yet in others the direction and magnitude of the contacting force may be needed. Matsushima et al. have experimented with an instrumented annular bumper with which the direction and magnitude of contact force can be determined [133]. The bumper appears as an elastically attached skirt around the robot. The resting position of the center of the bumper is known and compared with the new location after contact. Knowledge of the displacement of the center, points of attachment of the bumper to the robot, and the spring constants of the supporting members allows computation of the

force at the point of contact.

The interface between the robot gripper and the manipulanda is characterized by the force being applied by the fingers, slip of the object relative to the gripper, and the forces present at distant points in the arm resulting from manipulating the object. Yashikata reported on a gripper with strain gauge force sensors in the fingers which could be used to direct grasping [224]. Another facet of this information involves the principal forces [65] present at the gripper. These include the translational forces, along the x, y, and z axes, and the rotational forces about these same axes. A typical approach to measuring these forces is to construct a sensing element which fits between the gripper and the wrist of the robot or between a fixture and the work table [65], [213]. Specific examples of such devices are the instrumented remote center compliance device, or IRCC, cf. the wrist developed by Seltzer [187] and the Force Sensing Wrist developed by Robot Technology Inc. [183]. These devices are instrumented with strain gauges which provide information regarding the deformation of the structure. Similar in basic concept is the device developed by Folchi [67]. Proper design of the unit allows determination of all six force vectors. Such devices have been used to successfully guide the mating of close-tollerance parts [82], [83] [194]. The subject of slip has been the target of several ingenious mechanical sensors [202] also [95] wherein instrumented needles, wheels, or tracks in the fingers are used to detect movement of the object relative to the gripper but nothing is generally available. The magnetoelastic array sensor approach discussed by Hackwood [86] may, however, lead to an device capable of providing an "image" of the forces tangential to the sensor pad. This would represent a step toward the ability to detect incipient slip.

The forces present on the individual fingers of a gripper can also be determined. The IBM RS1 manipulator can be configured with a two-fingered, parallel-jaw gripper. The fingers are instrumented with strain gauges so that the pinching, side, and end forces can be transduced [1]. The gripper need not have "fingers" which resemble those of humans. In work done by Trevelyan [200], [199] in Australia, the end effector assumes the form of a wool shearing tool, see also [218]. The force being applied by the tool is used to modify the trajectory of the tool during the shearing process. Surface patches are used to model the shape of a sheep with the force feedback from the shearing head coupled with its position used to update the model for a given sheep.

1.4 Tactile Sensors

The final class of tactile sensors can be divided into three general categories depending upon the type of information they provide. Contact sensors compose the first class and yield information indicating either the presence or absence of physical stimuli. A robot hand with contact sensing on several sides was developed by Goto [84] to handle blocks placed on a table. In later work, Garrison and Wang [69] constructed a hand containing an array of contact sensors. A high spatial resolution conductive rubber contact sensing array has been developed at the Artificial Intelligence Laboratory at M.I.T. and has been

successfully used to identify parts from a small class learned by the system [98]. An interesting use of contact sensing has been reported by Hirose and Umetani [99]. Binary touch sensing on each segment of an active cord, or snake, mechanism was used to guide the motion of the device.

A manipulation system equipped with tactile sensors has been the subject of experimental work at Hitachi [191]. The purpose of the system was to test the ability to automate the process of containerized cargo handling. Contact sensors and angle sensors on the touch pads were used to provide the control system with positional information pertaining to the container being manipulated. The availability of this information contributed to achieving system performance on the same level as that attained by trained human operators.

The second category of tactile sensors contains the sensors which provide information regarding the magnitude of the force at each sensing point. [171] contains a review of some of the recent sensor designs. A hand constructed by Hill and Sword [97] employed analog force sensing arrays. Recent sensors have utilized woven graphite fibers as the physical transducer [117], [118]. A more recent device utilizes the Hall effect to transduce the positions of a set of tactile probes [113]. Hall cells are moved by contact with the stimulus object thus changing the distance between the cells and magnets fixed to the gripper. This change is measured via the Hall cells and made available to the controlling computer.

The "artificial skin" tactile array sensor developed by Clot and Briot at the Laboratory for Automation and Systems Analysis in Toulouse utilizes pressure variable transverse resistance to transduce the forces present on the pad [44], [45] [34]. This sensor is very similar in design to the device developed at JPL [38], [22] and was originally developed to assist in the analysis of the pressure distribution on the sole of the foot during walking but has sparked great interest in the field of robotics. The device consists of a printed circuit board on which is located an array of "sensitive points". The device used by Briot in the robot experiments consisted of 100 points arranged in a 10x10 array with 1 cm spacing between the centers of adjacent transducers. These transducers consist of a center measurement electrode surrounded by an annular electrode. A homogeneous pad composed of a conductive elastomer 5mm thick is placed on top of the contact array. A reference voltage is applied to the annular electrode and the current entering the center electrode is measured. Since this current is proportional to the resistance through the pad, it is an indication of the force present in that area.

A configuration very similar to the Clot and Briot sensor has been utilized by Becjzy and others at JPL [38]. Their scanning technique involves holding the annular guard rings at ground potential and connecting the center contacts to a reference voltage through a resistor. The current through the circuit is used as an indication of the force being applied.

Wolfeld at the University of Pennsylvania has experimented with a prototype tactile sensor provided by the Lord Corporation [217]. The sensor was reported to contain 64 transducers uniformly spaced in a square array roughly

2.1 inches on a side [177]

Recently, another company, Barry Wright, has entered the field of tactile sensing with a prototype tactile array sensor. Their device contains an array of 256 transducers arranged in a square with a distribution of 100 elements per square inch [184]. The device includes scanning hardware and provides the host computer with 8-bit digital values for the force elements. This device uses conductive rubber as the basis for the transducer and is similar in design to the sensor constructed by Hillis [98] and also the sensor of Purbrick [173].

Recent work by Raibert and Tanner [175] has been directed toward the development of a tactile array sensor employing VLSI technology. The intent is to place processing capabilities at each forcel in the array thus distributing the computation. This lends itself to the parallel, local types of computations typically done in image processing, cf. Chapters 3, 4, and 5. The largest functional sensor contains 18 forcels in a 6x3 array. A major disadvantage of this technique is the inherently planar nature of the sensors due to the structure of the underlying processing chip as imposed by the VLSI technology.

In another recent effort [19], [50], piezoelectric elements are used as the basic transduction device. The elements are placed in opposition to one another and separated by a compliant filler. The element closest to the surface of the sensor is used to produce an acoustic wave in the filler which is received by the second element. The time of flight is proportional to the separation of the elements which, in turn, is proportional to the force applied by the stimulus since force leads to compression of the filler.

The approach taken by Hackwood and his colleagues at Bell labs involves embedding magnetic dipoles in a compliant medium [86]. The sensing elements appear as semiconductor magnetic sensors employing either magnetoresistive or the Hall effect. These tactile sensors can transduce not only the normal forces on the pad but the tangential forces and torques as well. This is an emerging approach and no practical devices are presently available.

The final class of tactile array sensors includes sensors which are designed to transduce the three dimensional shape of an object. In one example application, a linear array of spring mounted probes was used to return the contour of the surface. Probe height was transduced by strain gauges mounted on the springs. The surface information was used to direct the subsequent grinding pass [3]. The tactile sensor was not used in a direct feedback loop during the actual grinding. Page and Pugh [159] presented a sensor which builds a contour map of the object being sensed as the sensor is lowered onto the object. Takeda constructed a hand with arrays of free floating needles as sensors. The needles would conform to the shape of the surface being touched, thereby providing identification information [193]. A similar tactile sensor has been presented by Sato et al. [185]. The sensor consists of an array of pins which, when lowered onto an object, provide the height of the points with which they contact. The height information is provided as a set of binary images representing different heights. The major disadvantage with this design involves the binding of the sensing pins due to side forces. While the authors present an analysis of

this phenomenon and suggest some compensatory measures, the problem is inherent in the design and severely limits its utility. Another problem lies in the physical size of the transducers. The overall sensor is rather large relative to the density of transducers and the minimum transducer spacing is greater than that of the sensor presented here.

Tactile sensing as defined here is a contact sense which provides data regarding the interaction between the sensor surface and a stimulus. Two basic questions arise at this point: what information is available and how is that information used? Expanding this thought, what information can be derived from the sensor data and what processing techniques can be employed to extract that information? Secondly, how is that sensor-relative information related to the world and used in high-level control structures?

The questions of tactile image processing are addressed in chapters 4 and 5 where both static and dynamic processing techniques are considered. These algorithms extract information from the data which relates the stimulus object, or features of the object, to the sensor. This sensor-relative information must be utilized by the control structure of the system. Chapter 6 discusses a conceptual structure which allows the integration of multisensory information and motion control into a high-level planning and control paradigm to produce coordinated activity. A question not explicitly addressed in detail concerns the bridge between the sensor-relative information and the world. This includes both representation of the processed information and the process by which the information is related to the world. The area of representation is addressed implicitly in Chapters 4, 5, and 6 and will not be considered further here. Relating the sensor information to the world is relatively straightforward and is dealt with briefly below.

Two scenarios for the application of a tactile array sensor can be envisioned. In the simplest case, the device is affixed rigidly to the workspace of a robot. That is, it is attached to the world coordinate frame of the system. The other scenario involves the use of the sensor on the end effector of a robot manipulator. The difference between the two cases appears as the fact that in the former the relationship between the sensor and world is fixed and objects and the movements of objects occur relative to that fixed coordinate frame and in the latter, the relationship between the sensor and the world is variable. All combinations of multiple sensors and arms can be considered as examples of these two cases.

A robot mechanism consists, abstractly, of a sequence of links and joints with one link attached to a fixed coordinate frame and the link at the far end of the chain comprising the gripper. The latter scenario includes a sensor rigidly attached to the gripper. The first case results from the degenerate case of a manipulator with no links or joints. The question is then one of describing the relationship between the coordinate system of the sensor and the base, or world, coordinate system.

In order to describe and discuss this relationship, one needs to adopt a notation and a set of conventions. In the Notation of Paul [160], homogeneous

transformations are used and the position and orientation of the end effector relative to the base of the robot is given as a matrix product of a set of A matrices. Each A matrix specifies the transformation defined by a link/joint pair. Another transformation matrix is used to describe the transformation between the world coordinate system and the coordinate system of the base of the robot. Yet another transformation matrix is used to describe the relationship of the sensor coordinate system to the robot end effector coordinate system. The product of these matrices provides a mathematical description of the relationship between the sensor coordinate system and the world coordinate system and allows points and vectors in one system to be described in the other. Thus sensor-relative information derived from tactile image processing can easily be related to the robot and the world from the standpoint of position and orientation.

This chapter presented a general discussion of sensory data and sensors specific to the robot domain. The intent was to provide insight into how tactile array sensing relates to other sensing capabilites. A classification scheme for sensory data was developed as was a taxonomy for sensing devices. From these it is seen that tactile sensing provides *DIRECT*, *EXTERNAL*, *LOCAL* information which may be either *STATIC* or *DYNAMIC*. The transducer itself is a *CONTACT* device and, depending upon how it is utilized, can be either *PASSIVE* or *INVESTIGATIVE*. Emphasis was placed on tactile sensing for obvious reasons and the particular area of interest in this work involves tactile array sensing. Tactile array sensing is defined as the process of acquiring and processing an array of numbers, a tactile image, produced by the array sensor which is indicative of the distribution of forces or deflections present at the interface between the transducer and the stimulus. The discussion will now focus on the design and behavior of various tactile array sensor.

Chapter 2

The Tactile Sensor

2.1 Introduction

The topic of this thesis is the acquisition, processing, and use of tactile, or touch, information in the identification of objects by machines. While there is increasing effort in the area of transducer design [95], a suitable tactile array sensor is not available. To date, tactile array sensors as distinct from devices like contact switches have been laboratory curiosities and have suffered from the problems of poor spatial resolution, poor electrical response (e.g. small and/or non-linear signals), hysteresis and fatigue, and short useful life. The need for a sensor prompted the development of the array sensor discussed in this chapter.

A tactile array sensor has been defined as a device which provides a matrix of numbers representing either the force of deflection distribution at the interface between the stimulus and the transducer. The basic component of any sensor is the transducer. That is, what device converts the phenomenon of interest into a signal which can be acquired by a machine? The answer to this general question contains two parts: identification of the quantity of interest and selection of the physical phenomenon used to measure that quantity. In the case of the this tactile array sensor, the quantity of interest is the force applied to or the deflection of the surface of the transducer and the physical phenomenon used is a change in the electrical resistance of a volume of material due to that force or deflection.

The "ideal" tactile sensor would exhibit several desirable characteristics [94]. These would include physical ruggedness, reasonable fabrication cost, high spatial resolution, low hysteresis, high sensitivity, large dynamic range, and monotonic response. The need for the first characteristic is obvious since a tactile sensor, being a contact sensor, must interact with the objects being sensed. Grasping of objects by robots consists of what essentially amounts to a collision between the manipulator and the object with unwanted forces. Since the sensor resides at the interface between the manipulator and the manipulanda, it must be capable of withstanding these forces. Ruggedness here includes resistance to environmental conditions such as extremes in temperature as well as physical construction. Since tactile sensors are used on robot grippers with the possibility of myriad applications, it is desirable to be able to fabricate them in arbitrary sizes and shapes and at a reasonable cost. The need for a reasonable fabrication cost stems from the economics involved in the application of robots. Some applications require very high spatial resolution. The ability to vary the spatial resolution across the sensor surface as well as to build high density arrays is needed. In the area of the response characteristics of the device, it should have a large dynamic range, low sensitivity, low hysteresis, and monotonic response within its usable range. It should also be capable of withstanding overload forces with no lasting effects. The work reported here is not intended to result in a product and therefore the issue of cost is not considered. Questions regarding

the needed spatial and force resolution are difficult to answer due to a general lack of understanding of how tactile sensing will be utilized. The remaining ares as they pertain to the sensor used in this work are discussed in a subsequent chapter.

A number of technologies have been considered for use as force array sensors. These include: conductive elastomers, capacitive, optical, piezoelectric, various mechanical, as well as others. Section 2.4 of Chapter 2 provides a brief overview of current approaches to tactile sensor design and a comprehensive review of current touch sensor technologies is given in [95]. A conductive elastomer is basically a rubber of some form filled with a conductive element such as carbon, silver, or nickel. The sensor used here employs a carbon-filled silicone rubber conductive elastomer. In a capacitive sensor, two plates are separated by an elastic insulator. The plates thus form a capacitor, with the value of the capacitance partially dependent upon the separation of the plates. As the force applied to the plates increases, the elastic separator compresses, this reducing the capacitance of the device. This change can be used as a indication of the applied force [73]. Several approaches using piezo- and pyroelectric polymers have been tried with the most promising approach involving the use of two piezoelectric elements separated by a compliant filler [19] [50]. One element is used as the transmitter of an acoustic wave and the other as a receiver. The time of flight between the elements is a function of the characteristics of the filler and the element separation. A force applied by a stimulus object compresses the filler thus reducing the time of flight. This time can be measured and used as an indication of the stimulus. The typical application of optical techniques involves a change in either the reflectance or transmission of a reference light source depending upon the applied force. The forces applied by a stimulus to a transducer produce stresses inside the transducer and these can be used to generate voltages in piezoelectric ceramics. Each of these approaches has advantages as well as disadvantages. For example, some conductive elastomers exhibit large amounts of hysteresis and capacitive sensors must be designed very carefully to avoid the possibility of a stimulus contributing to the capacitance of the transducers.

Two basic approaches can be taken to the conceptual design of array sensors. In the first approach, each sensor in the array is a discrete device with a dedicated path between it and the transduction hardware. For example, a simple visual imager could be constructed as an array of photocells each with a dedicated analog to digital converter. The other basic approach involves having the array of sensors share control and signal lines. These lines are in some way scanned and the sensor signals converted sequentially. Of course it is possible to develop any number of hybrid schemes; however, the point is one of differentiating between devices which are actually collections of discrete sensors and those which are integrated arrays. The device developed here is a scanned array in which the control and signal lines are shared among the individual sensitive sites; the discussion below concentrates on this type of device.

The previous chapter provided an overview of the various sensing

modalities and associated sensors available to robots. In addition, a detailed discussion of current sensors was given with emphasis on the relationships between those sensors and the one developed here. The device used here employs a scanning approach and an analysis of various array scanning circuits is provided in section 2.2. The generic design of the array sensor used in this thesis is presented in section 2.3. Section 2.4 contains details of the physical design and fabrication of the various versions. The response characteristics of the latest version are discussed in Chapter 3 along with algorithms which provide tactile image "correction".

2.2 Analysis of Scanned Array Sensors

As indicated by Larcombe [118] and Hillis [98], sensing of a single pressure point is possible in an array sensor with shared control and signal lines: a "scanned" sensor. However, complex force patterns are difficult to understand due to the electrical interconnection of the individual force-detecting elements. This problem is alleviated in the present design by the insertion of a diode in series with each of the elements.

As discussed above, a tactile array sensor is composed of a collection of transducers arranged in some convenient spatial organization with the transducers providing signals correlated to force or deflection which are amenable to measurement by an electronic circuit. The transducers may be connected to the measurement circuitry in two basic ways. Each individual transducer can be connected directly to the peripheral circuitry. While this configuration is the easiest with which to deal on an electronic basis, it utilizes two wires, or paths, per transducer. A simplification of this design involves allowing all of the transducers to share one lead. This configuration requires n+1 lines where n is the number of transducers.

A more common method consists of defining two orthogonal axes on the transducer array and scanning these axes to determine the value of the individual transducers. One line is utilized for each row and column in the array. The scanning process produces a raster of force values called a "force image." The values returned are correlated to the resistance across the force detector connected to the selected digital line. Since the resistance is proportional to compression of the element, the value returned is an indication of the force being experienced at that detector.

The basic problem is how to collect the readings from the force transducers in a spatially organized array. The array sensor developed in this work utilizes pressure-variable resistors as the transducers. The conceptual organization of an NxM array of transducers connected so as to allow scanning along two orthogonal axes is given in Figure 2.1.

The following section discusses various techniques for measuring the variable resistances and the problems associated with each technique. The discussions focus on the analysis of a scanned sensor with two rows and two columns of pressure-variable resistance transducers. The techniques can be extended to the general NxM case in a straightforward manner, but the equa-

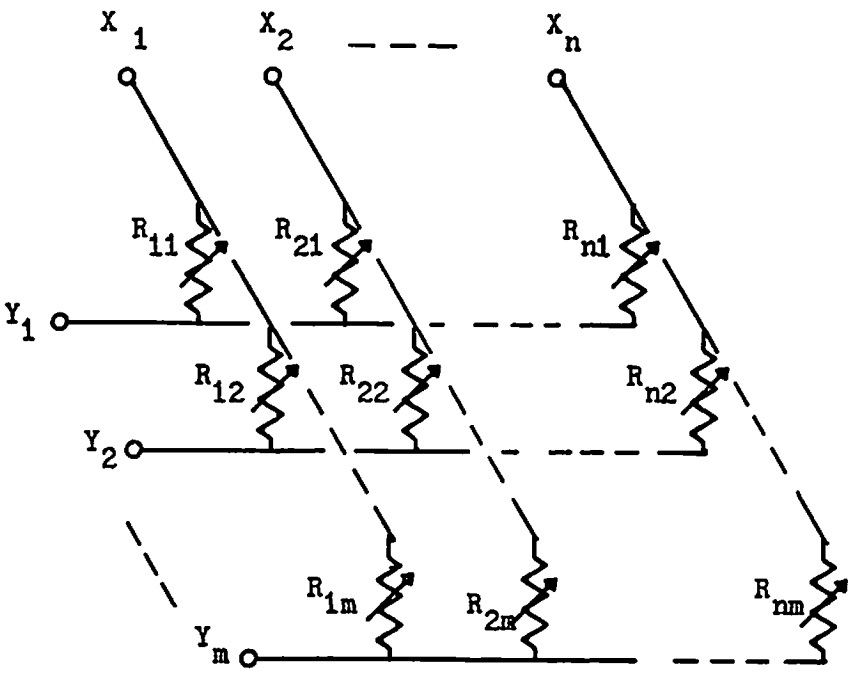


Figure 2.1

Conceptual organization of an NxM array of transducers. The transducers are variable resistors and the basic configuration uses shared lines. A given transducer is accessed via its x and y line address.

tions become burdensome. The 2x2 case contains all of the elements essential to elucidate the points of interest in the various techniques.

2.2.1 Case I: Floating Line

A typical method for measuring the values of the unknown resistors is to measure the potential across a dropping resistor, i.e. a resistor of known value connected to ground and through which passes a current proportional to the resistance to be measured. The dropping resistors are placed in series with the resistances to be measured. In the ideal case, the circuit is simply a voltage divider; thus, the value of the unknown resistance is directly related to the potential across the dropping resistor and the dropping resistor's value. The schematic for 2x2 sensor with dropping resistors is given in Figure 2.2. The two axes are denoted by x and y in the figure. The location of a particular sensor is provided by the x and y lines to which it is attached.

The process utilized to measure the value of the transducer at location Xb,Yd in the array involves holding the B x line in Figure 2.2 at some reference voltage while allowing the remaining x line, A, to float and then measuring the potential between the Yd y line and ground. This potential is the voltage

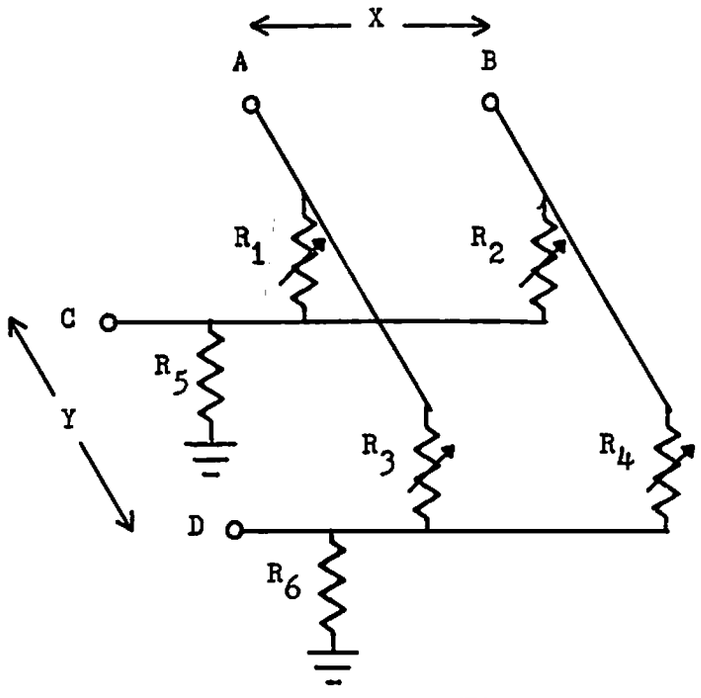


Figure 2.2

Schematic of 2x2 array sensor with dropping resistors. The dropping resistors provide a reference to which the values of the variable resistor are compared. A particular element in the array is selected by holding its X line high and sensing the voltage on its Y line.

across the dropping resistor connected between the Yd line and ground. Ideally, the voltage sensed would be a function of only the single transducer of interest, Rbd, the reference voltage, and the dropping resistor value. However, the transducers are not electrically isolated and voltage measured is actually a function of all of the transducers in the network. The equivalent circuit for the circuit shown in Figure 2.2 under the previously mentioned scanning conditions is given in Figure 2.3a.

Substituting Rb=R1+R3 yields the circuit given in Figure 2.3b which, in the 2x2 case, is the well known Wheatstone bridge. The "In" terms indicate current flow in the steady state and define the quantities used in the following analysis. Using Kirchhoff's circuit laws gives the following current equations for nodes B,D, and E in Figure 2.3b:

node B:	I0 = I2 + I4	or	I2 + I4 - I0 = 0	(1)
" D:	I4 = Ib + I6		Ib + I6 - I4 = 0	(2)
node E:	I5 + I6 = I0		I5 + I6 - I0 = 0	(3)

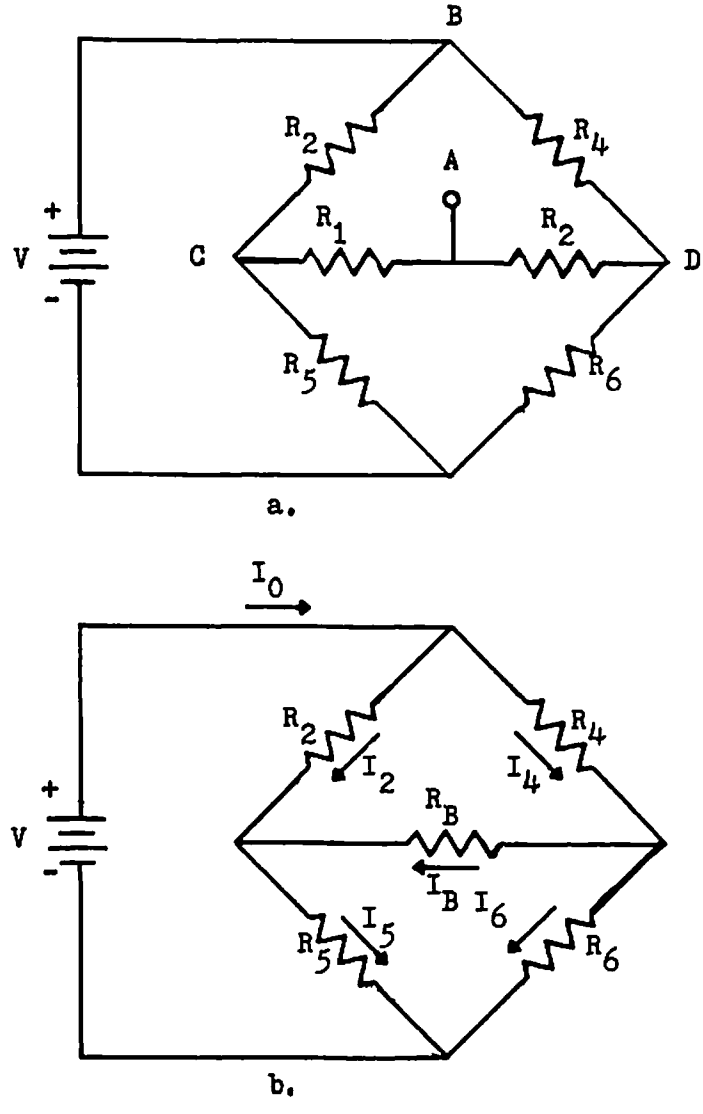


Figure 2.3

2.3a: Equivalent circuit for the array sensor in Figure 2.2 when resistor R4 is of interest. X line B is held high, x line A is allowed to float. 2.3b: current paths through circuit in 2.3a with resistors R1 and R3 lumped into Rb.

The voltage equations are:

$$\begin{aligned} &\text{path B, D, E:} && V - I4R4 - I6R6 = 0 \\ &\text{"} \quad \text{B, C, E:} && V - I2R2 - I5R5 = 0 \\ &\text{path B, C, D:} && I2R2 - IbRb - I4R4 = 0 \end{aligned}$$

Rewritten, these yield:

$$\begin{aligned} R4I4 + R6I6 &= V && (4) \\ R2I2 + R5I5 &= V && (5) \\ R2I2 - RbIb - R4I4 &= 0 && (6) \end{aligned}$$

We need to solve for the currents I5 and I6 since the voltages at points C and D relative to E (ground) are simply the products of these currents and the dropping resistor values. This system of equations can be solved if they can be rewritten in terms of only three variables. In order to simplify to only three variables, two currents from the set Ib,I2,I4 must be eliminated from equations (4),(5), and (6). In this case, I2 and I4 are eliminated

$$\begin{aligned} &\text{from (2):} && I4 = Ib + I6 && (7) \\ &\text{from (1):} && I2 = I0 - I4 && (7b) \\ &\text{from (3):} && I0 = I5 + I6 && (9) \\ &\text{from (7),(7b), and (9):} && I2 = I5 - Ib && (8) \end{aligned}$$

$$\begin{aligned} &\text{from (4) and (7):} && R4Ib + (R4+R6)I6 = V && (10) \\ &\text{"} \quad \text{(5) and (8):} && -R2Ib + (R2+R5)I5 = V && (11) \\ &\text{from (6),(7),(8):} && -(R2+Rb+R4)Ib + R2I5 - R4I6 = 0 && (12) \end{aligned}$$

Written in matrix form, this set of equations becomes:

$$D \begin{vmatrix} Ib \\ I5 \\ I6 \end{vmatrix} = \begin{vmatrix} V \\ V \\ 0 \end{vmatrix}$$

Where the coefficient matrix, D, is given by:

$$D = \begin{vmatrix} R4 & 0 & (R4+R6) \\ -R2 & (R2+R5) & 0 \\ -(R2+Rb+R4) & R2 & -R4 \end{vmatrix} \qquad (Q1)$$

and the solution is then:

$$\begin{vmatrix} Ib \\ I5 \\ I6 \end{vmatrix} = D^{-1} \begin{vmatrix} V \\ V \\ 0 \end{vmatrix} \qquad (Q2)$$

where D^{-1} is the inverse of the coefficient matrix given by:

$$D^{-1} = \frac{1}{|D|} \begin{vmatrix} -R4(R2+R5) & R2(R4+R6) & -(R4+R6)(R2+R5) \\ -R2R4 & R4(R2+Rb+R6)+R6(R2+Rb) & -R2(R4+R6) \\ R2(R4+R5+Rb)+(R5(R4+Rb) & -R2R4 & R4(R2+R5) \end{vmatrix} \qquad (Q3)$$

The inverse of D is defined if its determinant, |D|, is non-zero. The determinant of the coefficient matrix is:

$$|D| = (R2R4+R4R5+R2R6+R5R6)Rb + (R2R5+R2R6+R5R6)R4 + R2R5R6 \quad (12b)$$

By definition of the circuit, R5 and R6 are non-zero. Also, all resistances are positive. From equation 12b, the determinant is zero only if all of the variable resistors, R1, R2, R3, and R4, are zero.

The voltage with respect to ground at point D, across resistor R6 in Figure 2.3a is:

$$VD = R6I6 \quad (13)$$

Thus current I6 is given from equation Q2 as:

$$I6 = V \frac{R2R5 + R2Rb + R4R5 + R5Rb}{(R2R4+R4R5+R2R6+R5R6)Rb + (R2R5+R2R6+R5R6)R4 + R2R5R6} \quad (16)$$

This discussion has focused on the analysis of a Wheatstone bridge. This type of bridge circuit is typically used to measure the resistance of a transducer. In such applications the transducer appears as the resistor in one of the legs of the bridge, R2,R4,R5, or R6 in Figure 2.3b. A current sensor would monitor the current, Ib, through the bridge resistor, Rb. One of the three remaining "leg" resistors is varied until the bridge current is zero. With Ib=0, the voltage at point C is equal to that at point D and the value of the unknown resistance is easily calculated from the values of the three known resistors using the relationship:

$$\frac{R2}{R5} = \frac{R4}{R6} \quad (17)$$

A 2x2 scanned sensor is equivalent to such a circuit except for two points. First, the bridge resistor, Rb, is actually the series combination of the two transducers connected between the floating x line and the two y lines, i.e. Rb=R1+R3. In addition, where in the Wheatstone bridge the values of the leg resistances are known, the corresponding resistance values in the sensor, as well as the bridge resistor resistances, are unknown. That is, R1, R2, R3, and R4 are unknown.

The approach taken here to transduce the force and (x,y) location (B,D) in the 2x2 array has been to measure the voltage with respect to ground at y line D. This voltage is given by equation (16). The important point to note is that the voltage is not only a function of resistor R4 as desired, but has the undesirable property of being dependent upon the values of R1, R2, and R3 as well. The values of these three resistors are functions of the forces present on the sensor and are the quantities to be measured, along with R4.

The point of this exercise is to understand the relationships of the various resistances to the voltage seen at point D. The interesting feature is that the voltage is a function of all of the resistors in the network and in fact, is non-linear in some variables. In the case of the array sensor as shown in Figure 2.2, the dropping resistors, R5 and R6, can be set equal. The values of the other four resistors are indicative of the forces being applied to the sensor at their respective locations and are the quantities to be determined.

Equation (16) expresses the current through the dropping resistor R6 in terms of the reference voltage and the resistances in the network, i.e. the sensor. Similar expressions can be determined for the voltage at point C, and again for points D and C but with x line A held high and B allowed to float. Given these four equations, the observed voltages, and the values of the dropping resistors, the values of the four unknown resistors can be determined.

The above discussion relates to a 2x2 sensing array. This analysis may be generalized to an nxn array with an associated increase in complexity. Given the complexity of the 2x2 case, it is easy to see that an analytical solution for the nxm case would be difficult to compute in a real-time fashion.

2.2.2 Case II: Grounded Line

One method for simplifying the calculation outlined above is to force all of the x lines to ground when they are not being held high, i.e. if B is the x line to be sampled, A is connected to ground. In this case, the 2x2 sensor schematic illustrated in Figure 2.3a is equivalent to that shown in Figure 2.4. The analysis for the right path, BDA, proceeds as follows. The total resistance along that path is equal to the series combination of R4 and the parallel combination of resistors R3 and R6:

$$Rt = R4 + \frac{R3R6}{R3 + R6} \tag{18}$$

The total current in the branch is given by:

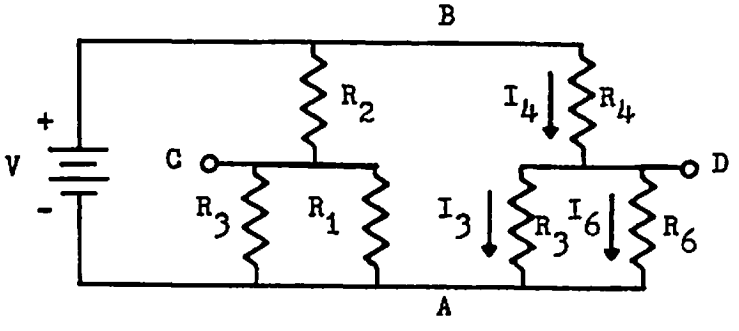
$$It = \frac{V}{Rt} = V \frac{R3 + R6}{R4(R3+R6) + R3R6} \tag{19}$$

and finally, the voltage seen at point D is simply:

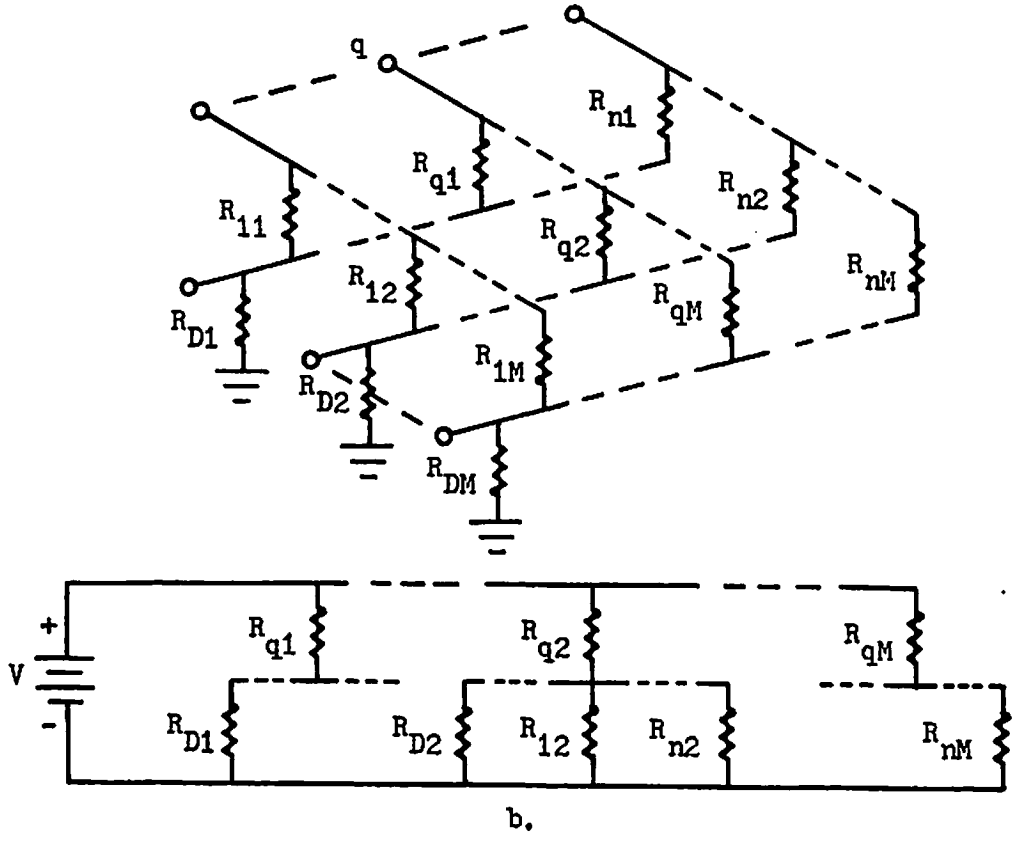
$$Vd = \frac{It\ R3\ R6}{R3 + R6} = \frac{V\ R3\ R6}{R4(R3+R6) + R3R6} \tag{20}$$

Figure 2.4

2.4a: Schematic of alternate scanning configuration for the circuit in Figure 2.1. In this case, the non-selected x line is forced to ground. In order to transduce resistor R4, line A is forced low while line B is held high. This causes the variable resistances of R1 and R3 to appear in parallel with the dropping resistors. 2.4b: equivalent circuit for the general case of the sensor shown in Figure 2.1 with x line "q" chosen.


B
R2
I4
R4
+
V
-
C
D
R3
R1
I3
R3
I6
R6
A

a.


q
Rn1
Rq1
Rn2
R11
Rq2
RnM
R12
RqM
RD1
R1M
RD2
RDM
+
V
-
Rq1
Rq2
RqM
RD1
RD2
R12
Rn2
RnM

b.

This circuit is simply a voltage divider. While the computation is far simpler than in CASE I, the voltage at the sensing point, point D, is still dependent upon the values of other transducers in the same y column. In this case, the (unknown) value of the transducer at location (A,D) enters the equation relating the voltage at point D with the resistance R4. When implemented for a general nxm sensor, the computation of the unknown resistances is again prohibitively large. Figure 2.4b illustrates the general mxn case with x line "q" chosen to be sampled. The raw voltage values recorded at the sensor must undergo a processing phase before a true "force image" is obtained.

2.2.3 Case III: Current Injection

We now describe the method employed by Hillis which is similar to the grounded case discussed in the previous section. To this point, the circuits considered have employed a dropping resistor and the potential measured across this resistor used to determine the value of the unknown resistor. The measurement of this type of circuit is passive. The circuit used by Hillis does not contain the dropping resistors, see Figure 2.4b, and uses an active technique for measurement.

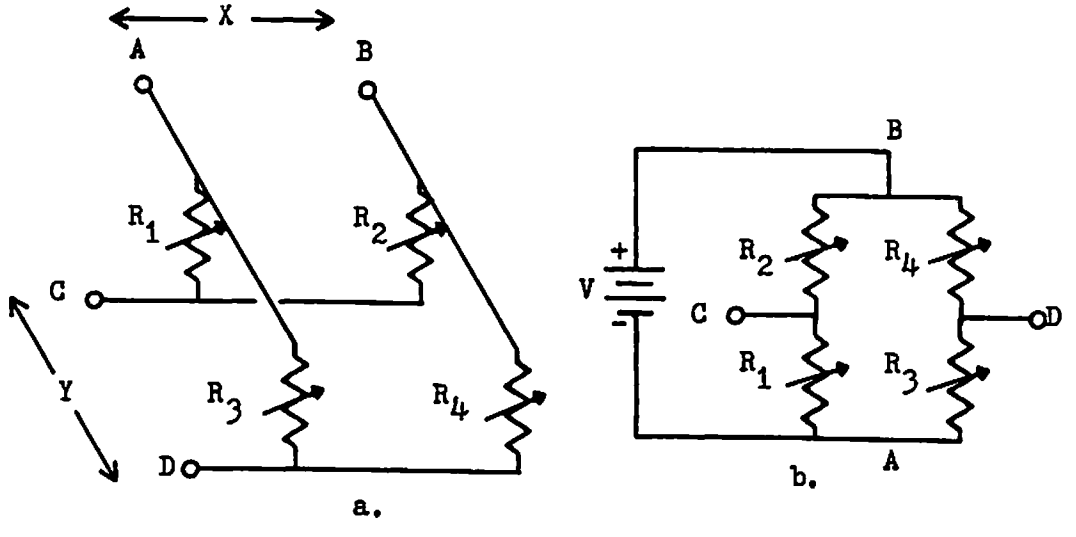


Figure 2.5

Equivalent circuit for the Hillis sensor. 2.5a: ideal generic case. 2.5b: scanning configuration again with x line b selected and x line a forced to ground. This causes the variable resistors to form a voltage divider. Current is injected into the y lines c and d in an amount required to bring these points to ground potential. The values of the resistors of interest can then be determined.

As in the previous cases, the x line corresponding to the transducer to be measured is held high while the remaining x lines are forced to ground in order to remove alternate paths from the circuit. The equivalent circuit is shown in Figure 2.5b.
The resistance of path from the selected x line to ground through a given y line, sense point, is the series combination of the resistance to be measured and the parallel combination of the resistances in the same y row.

Current is injected into each of the y lines in exactly the amount required to bring the line to ground potential. The value of the unknown resistor is then inversely proportional to the current injected and the crosstalk among the transducers is eliminated. For a complete development, see [98]. A very similar approach is taken by Purbrick and the reader is refered to [173] for details. In the example 2x2 transducer array, each path is a simple voltage divider and it is clear that the current needed to bring a given y line to ground is a function only of the resistance to be measured.

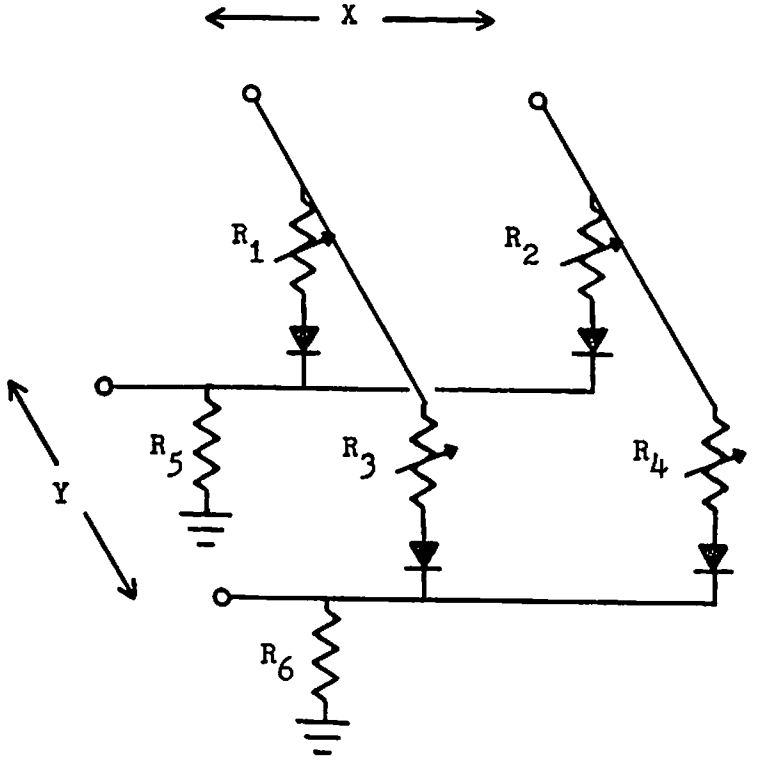


Figure 2.6

Schematic of 2x2 array sensor employing a diode in series with each transducer to eliminate electrical coupling. Dropping resistors are used as in Figure 2.2 as references.

2.2.4 Case IV: Diode Matrix

Case I above illustrated the difficulties in understanding the actual stimuli resulting in an array of voltages at the y line sensing points. Case II involved tying the unused x lines to ground. This technique still resulted in the need for a preprocessing stage before the values obtained at the sensor reflect the actual physical stimuli. Case III discussed a method for eliminating these difficulties by injecting current into the unused x lines with additional circuitry. An alternative approach similar to that utilized in keyboard technology can be used to completely isolate each transducer from the rest of the network.

A diode placed in series with each transducer allows current to flow in only one direction through the transducers. Furthermore, as in Case II, all unused x lines are forced to ground. This general configuration is shown schematically in Figure 2.6.

For the analysis, consider the case when trying to measure transducer BD. The x line B is held at some reference voltage V while line A is forced to ground. The equivalent circuit for the sensor in this configuration is shown in Figure 2.7.

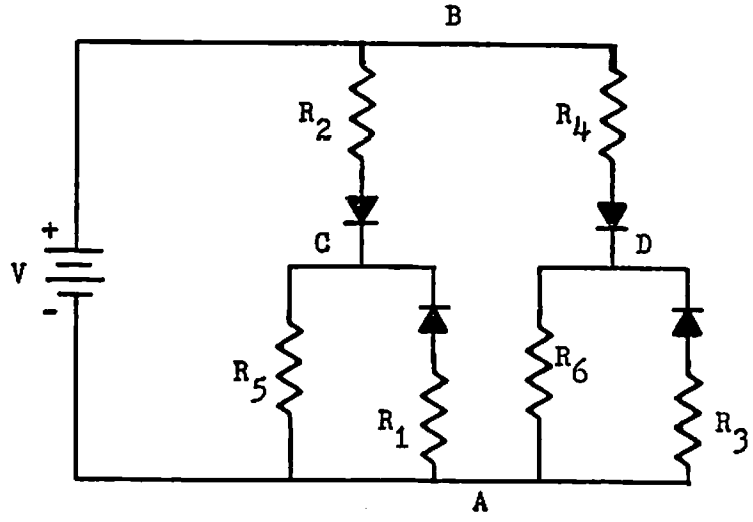


Figure 2.7

Equivalent circuit for the array sensor in Figure 2.6 under the same scanning conditions as in Figure 2.4. X line b is held high, x line A is held at ground. The diodes in series with resistors R1 and R3 are back biased thus removing the parallel combination effect seen in Figure 2.4

Since point B is at a higher potential than point A, by definition, points C and D must be either at the same potential or higher than A. This means that no current will flow through resistors R1 and R3 since the diodes are

reverse biased. The potential at point D is given by a simple divider relationship:

$$Vd = \frac{V\ R6}{R4 + R6} \qquad (21)$$

The important point is that Vd depends only on the reference voltage, the known, fixed dropping resistor value, and the variable resistor comprising the transducer. The relationship of Vd to the variable resistance is given in Figure 2.8.

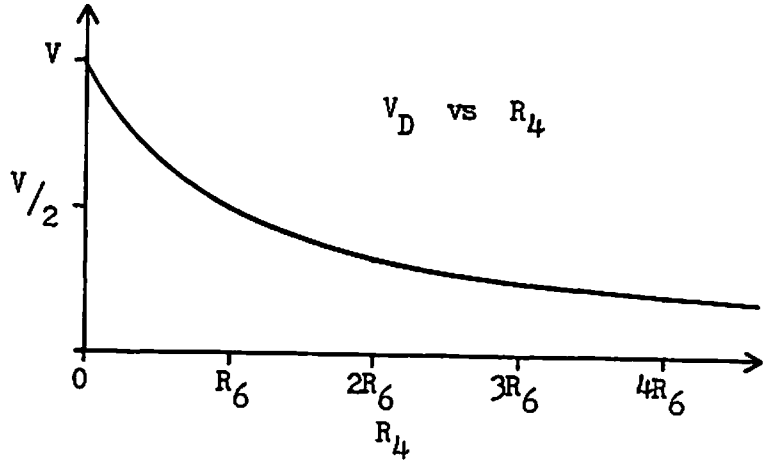


Figure 2.8

Vd vs. variable resistance of R4 calculated from equation 21. Vd is in terms of the reference voltage, V, the reference resistor, R6, which is fixed, and the variable resistor, R4.

The major drawback of this method is that the sensor array must either have one wire for each transducer plus one for each x line or the diodes must be fabricated into the sensor directly. The latter approach is taken here and is discussed in detail below.

2.3 Generic Sensor Design

This section discusses the design of the generic tactile sensor array utilized in this work. This design consists of three major components: the actual transducer array (or sensor pad), a printed circuit board (PCB), and the connections between the pad and the board. The sensor pad contains transducers whose response characteristics correlate to the force being applied to them. The interface between the pad and the scanning circuitry is provided by the PCB. One of the most difficult problems involves the connections between these components.

In the theoretical analysis given above, the transducers were modeled as

discrete resistors. In practice one must choose between at least three implementations of the transducer array. The first and simplest uses a single block of a conductive elastomer for the sensor pad. This material is typically composed of a silicone rubber doped with carbon particles. As the resulting material is compressed, the carbon particles are pressed together, increasing the contact area and thereby reducing the resistance. A more thorough treatment of the conduction mechanism apparent in filled elastomers is found in [195]. A set of parallel bare wires is affixed to the top of the pad while a second set aligned orthogonal to the first set is affixed to the bottom. This configuration is shown in Figure 2.9.

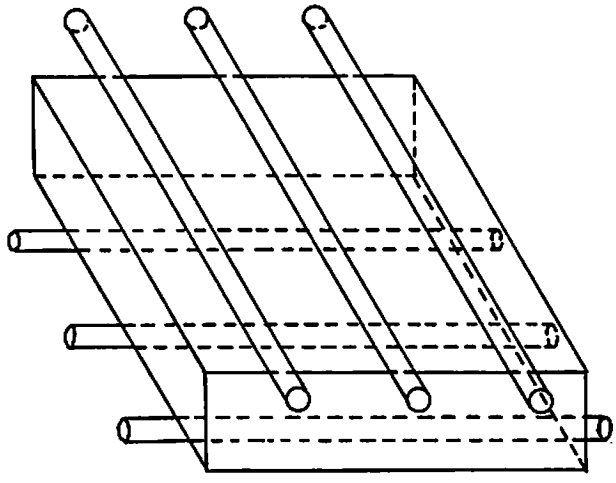


Figure 2.9

Physical configuration of the simplest approach to sensor construction. This design is described by the schematic in Figure 2.1 and thus contains no dropping resistors. The variable resistor matrix is implemented as a single block of carbon filled rubber. The x and y lines are formed by orthogonal sets of conductors.

While this design is relatively easy to fabricate, it does exhibit a limitation. In this case the sensor pad is actually a single, large, variable resistor. The resistance is measured between two lines on opposite sides of the pad. The transducer is not divided into spatially discrete sections: it covers the entire surface of the sensor. Forces present at distant points on the sensor vary the resistance in that local area. However, since the block is homogeneous, these variations affect the resistance measurements at other points on the block. Referring back to Figure 2.2, this condition causes the resistors R1,R2,R3, and R4 to be dependent upon one another. The analysis of case I above becomes

even more complicated.

A second possibility for pad construction consists of replacing the solid block and wire connectors by two sheets of striated conductive rubber as used in LCD connectors and flexible PC jumpers. The structure of this material is depicted in Figure 2.10.a. A sheet is composed of alternating layers of a conductive elastomer and an insulating rubber. As the two sheets are pressed together, the resistance at the intersection of any orthogonal pair of conducting bands decreases.

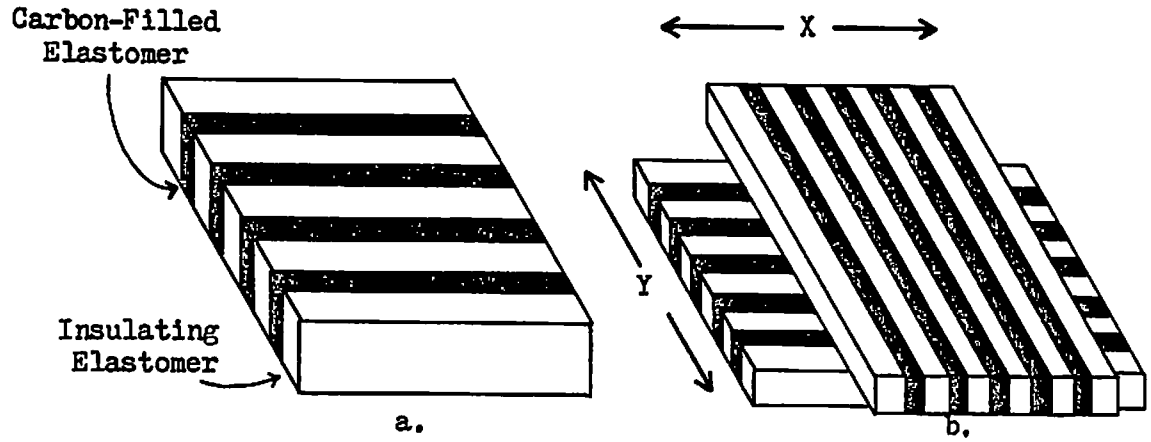


Figure 2.10

This shows an alternate structure for the sensor pad. 2.10a: this design uses sheets of striated rubber containing bands of carbon filled rubber separated from one another by bands of insulating rubber. 2.10b: two sheets of this material are laid on top of one another to form both the variable resistors and the x and y lines. The variable resistors appear as the variable contact resistances at the intersection of the lines.

This type of material can be purchased with varying numbers of conductors per unit width up to 240 per inch. A sensor pad is constructed by placing two such sheets in opposition as shown in Figure 2.10.b. The method employed by Hillis [98] is an extension of this approach.

The modification utilized by Hillis involves replacing the sheet of rubber forming the y lines in Figure 2.10.b with a flexible PC board. The resulting resistance along a particular y line is negligible compared with the quantities being measured and thus can be ignored. The resistance at the point of contact between the rubber sheet and the PCB decreases with increasing pressure. Recall that Hillis uses the current injection scanning method which does not require the dropping resistors. The equivalent circuit is given in Figure 2.10.b, cf. Figure

2.11. The quantities of interest in this design, R1,R2,R3, and R4, are the contact resistances between the conductive elastomer x paths and the PCB y lines. The greater the pressure applied to an x line - y line intersection, the lower the resistance. Hillis accentuates this property in practice placing a fine mesh between the layers as a separator. This separator serves the function of increasing the amount of force required to produce a given area of contact between the rubber sheet and the PCB.

Measurement of the forces present in the two-sheet design, and in the Hillis version, is obtained by sensing the resistances between the various pairs of x and y conductors. This design eliminates the influence of distant areas of the sensor to the measurements in local sections which was a problem in the homogeneous pad design. However, the design has one remaining problem.

Recall that in the first implementation the x and y connections were provided by wires. The resistance along the length of these wires is negligible compared to the resistances seen through the pad. In addition, the resistance per unit length does not change with increasing force. In the present configuration, cf. Figure 2.10b, the wires have been replaced by the conductive bands. In the ideal situation, the resistance along these strips can be relatively low. However, as pressure is applied the resistance changes. If the resistance, or voltage, measurement is taken at the one end of a y line or the reference volt-

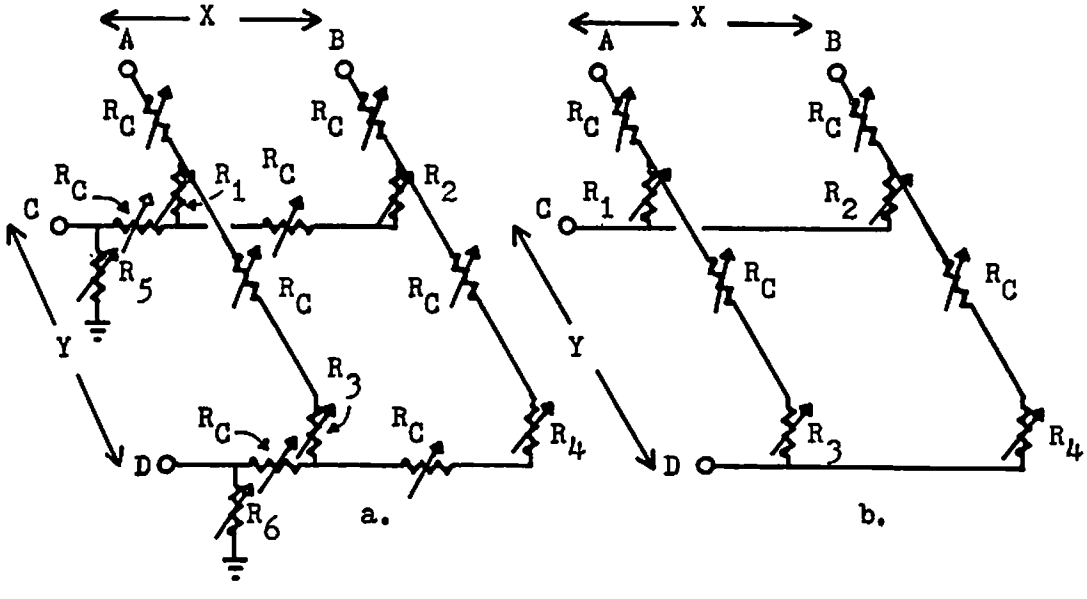


Figure 2.11

This contains the modified equivalent circuit for the Hillis sensor. The technique shown in Figure 2.10b results in variable resistors appearing in the equivalent circuit between each node.

age is applied at the end of an x line, then a force applied anywhere along either the x or y path will affect the reading. This situation is given schematically in Figure 2.11a with the Hillis version shown in Figure 2.11b, cf. Figure 2.2. Under normal conditions the contributions from the transverse resistances may be negligible compared with the quantities of interest but the basic design still functions as a bridge circuit with all of its associated complications. The major merit of this implementation of the sensor pad is its simplicity: this third method eliminates the problems cited in the previous discussions at the expense of increased fabrication difficulty.

As mentioned earlier, the ideal sensor pad would exhibit several features. One feature of interest is the spatial density of the transducers. The array would have spatial resolution high enough to allow discrimination capabilities on the order of those of humans. This is not to say that the transducer density needs to be equal to that of biological systems but only that it must be high enough to perform on a par with biological systems.

The effective resolution of a tactile sensing array can be increased in a rather straightforward manner. Since it is a contact sensor, the force image produced is in direct correspondence to the actual object. If one assumes that the individual transducers in the array sample the environment at a point rather than integrate over a small area, then resolution can be increased by simply moving the sensor relative to the environment. In actuality, the sensors do not sample at a point but rather integrate over a small area. If this area is small enough, however, it is reasonable to model the sampling as if it occurs at a point. Moving the sensor one half of the transducer spacing in the plane of the pad will increase the effective resolution by a factor of two. Repeating this procedure to form a square increases the resolution by a factor of four. This technique was used by Wolfeld [217] to try to increase the effective resolution of the Lord sensor. Each transducer would be spatially discrete and electrically decoupled from all other transducers in the array. This requirement of spatial discreteness implies that a transducer must have a small physical size compared with the entire pad, and that only forces present in the immediate area of the transducer affect its resistance. Electrically decoupling the transducers eliminates the affects of other transducers to the measured resistance of a particular element. Since the transducers are contained in a homogeneous block of material, there is some mechanical coupling. This coupling is discussed in a subsequent section and does not pose a problem as long as distinct objects yield distinguishable force/deflection images. Another characteristic involves the x and y connections. These connections should have no resistance and thus make no contribution to the resistance across the sensor.

The generic sensor pad utilized in the design discussed below consists of a spatial array of carbon-doped elastomeric columns embedded in a nonconductive body. Each column acts as a variable resistor whose response is correlated to compression, or force. This material has the property that its resistance decreases as it is compressed. Since the terms compression and force are related by essentially a scaling factor, they will be used interchangeably. The discrete

nature of the columns removes any influence from other transducers located elsewhere in the pad. The fact that the pad is a homogeneous block implies that there will be some mechanical interaction among neighboring elements. Consider a sponge: the area of the sponge surrounding a point of contact is also influenced by the contact. This type of interaction is common to all forms of array sensors.

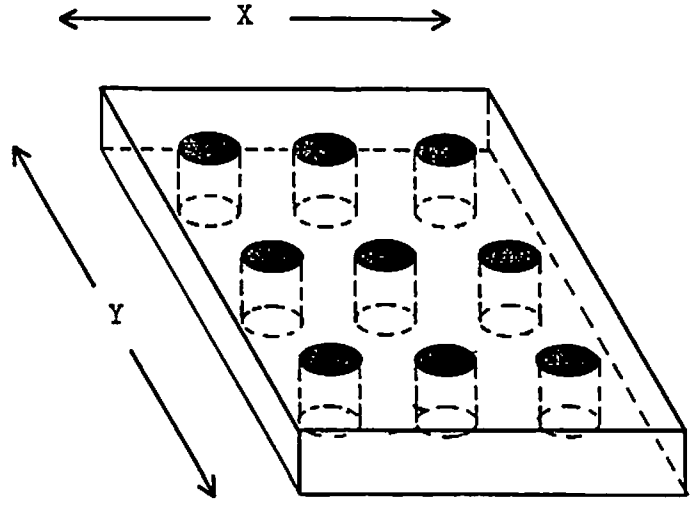


Figure 2.12

Alternate configuration for sensor pad. The transducers are implemented as conductive columns of carbon-filled rubber in an insulating block of silicone rubber. This eliminates electrical interaction between the transducers themselves.

The x line and y line connections are provided by wires. The resistance per unit length of the wires is negligible compared to resistance through a column in the pad, thus satisfying the final characteristic defined above. The most difficult characteristic to be dealt with involves the electrical decoupling of the transducers. The approach utilized is exactly as described in Case IV above. A diode placed in series with each transducer, or column, provides the decoupling. Figure 2.13 depicts the generic tactile array sensor design. The top layer of wires is held in place with a layer of tape. This design is similar that of Snyder and Clair [189] in that a diode matrix is used to electrically isolate the transducers. However, the sensor pad used by Snyder was a homogeneous block of conductive rubber while the pad used here employs a separate conductive zone for each transducer.

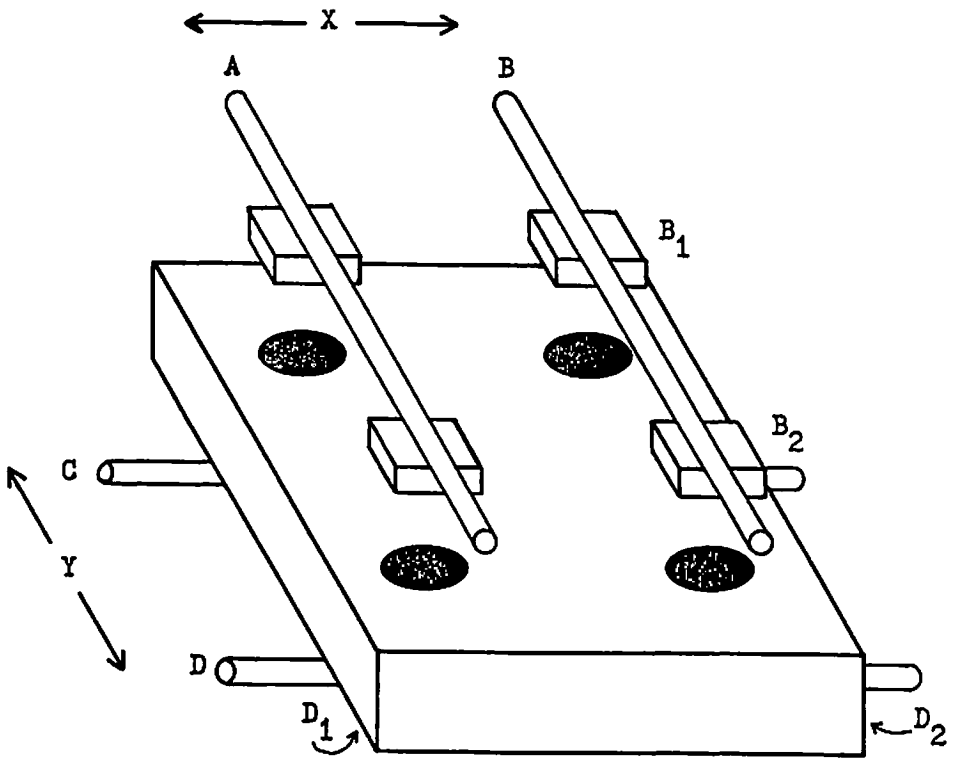


Figure 2.13

This illustrates the generic tactile array sensor design which employs a diode in series with each transducer element. The transducers are carbon-filled rubber cylinders as shown in Figure 2.12.

In the equivalent circuits presented in the analyses above, the transducers appeared as a single resistance. This resistance was intended to model the total resistance across the path from one x line to one y line, e.g. the path B1, B2, D2, D1 in Figure 2.13. From the previous discussion this resistance is actually seen as the sum of several separate components: Rb - the resistance along the conductor from B1 to B2, Rbd - the resistance across the transducer from B2 to D2, and Rd - the resistance of the lower conductor from D2 to D1, cf. Figure 2.13. Since the x and y lines are copper wires, Rb and Rd are negligible in this design and can be ignored, see previous discussion. Attention needs to be focused on the transducer resistance Rbd.

In practice, this resistance is also the sum of several components: Rcd - the resistance between the x conductor and the diode, Rd - the forward resistance of the diode, Rdt - the resistance between the diode and the carbon-doped column, Rt - the resistance of the column, and Rtc - the resistance between the

column and the y conductor. Rcd and Rd are negligible relative to Rt and are ignored. The remaining quantities Rdt, Rt, and Rtc are all dependent on the force being applied to the pad at their location. Rdt and Rtc represent contact resistances and Rt the resistance of the transducer itself. Thus the resistance being measured in this form of the sensor is a lumped quantity representing both the transducer resistance and the resistance between the transducer and the supporting circuitry. The relative contributions of the contact resistance to the transducer resistance is unknown but thought to be appreciable. Designing an experiment which measures only the contact resistance has proved to be very difficult and the actual contribution to the total variable resistance is unknown. A few researchers have attempted to separate the contact resistance changes from the internal resistance, "constrictive resistance", changes with differing conclusions. In the case of Tamai [195], a change in the resistance of carbon-filled rubber with compressive load is seen while the experiments by Piccirillo [165] show none. The quantity of interest here is the lumped resistance and, therefore, this point was not pursued.

One remaining question concerns the location of the diodes and the allocation of x versus y lines. The diodes could be located off-sensor. This would remove any size constraint on the diode matrix but would require one wire connecting each transducer to the corresponding diode. Such a configuration with n transducers and m x lines would require a total of n+m wires connecting the sensor pad to the scanning circuitry. Since it is intended that this tactile array sensor be affixed to the end effector of a robot, it is not only desirable to have high transducer density with as little additional volume as possible but also to reduce the number of connections, or wires, running between the sensor and the scanning circuitry.

The present design does not employ any scanning or processing circuitry at the sensor. This approach has been taken by Raibert [175] and also by the Lord Corporation [177] and is indicative of a growing school of thought. Distributing the tasks of data acquisition and preprocessing to the sensor, thereby producing an "intelligent sensor," is probably the direction in which successive generations of robot control architectures will move. While the author advocates this approach, the present work is concerned with the tactile sense as an information channel available to the robot's control system, whatever its form, and not with the details of the architecture of the controller. Thus questions surrounding how functionality is allocated between a tactile sensor subsystem and a supervisory computer are not considered. In the current design all of the x and y lines are brought off the sensor as fine wires to remotely located scanning/processing hardware.

Returning to the question at hand, the diode matrix is built directly into this sensor. Thus an array with k transducers employing n x lines and m y lines is described by the following equations:

$$k = nm \qquad \text{(A1)}$$

where n and m may assume only integer values and

$$f(n,m) = n + m \qquad (A2)$$

defines the total number of lines required. Combining (A1) and (A2) yields:

$$f(n) = n + k/n \qquad (A3)$$

with

$$f'(n) = 1 - k/nn \qquad (A4)$$

The derivative of f(n) given by equation (A4) has exactly one zero corresponding to the minimum of f(n) which occurs at n=SQRT(k). As one would expect, and from equation (A1), n=SQRT(k) which produces a square array. This indicates that the minimum total number of wires needed for an orthogonal axis scanning sensor is 2SQRT(k) where, again, n is the number of transducers in the sensor.

2.4 Sensor Development

Discussed below are the various versions of the tactile array sensor that we constructed. Details of the construction of sensor pads, the underlying circuit boards, and the connection techniques are given.

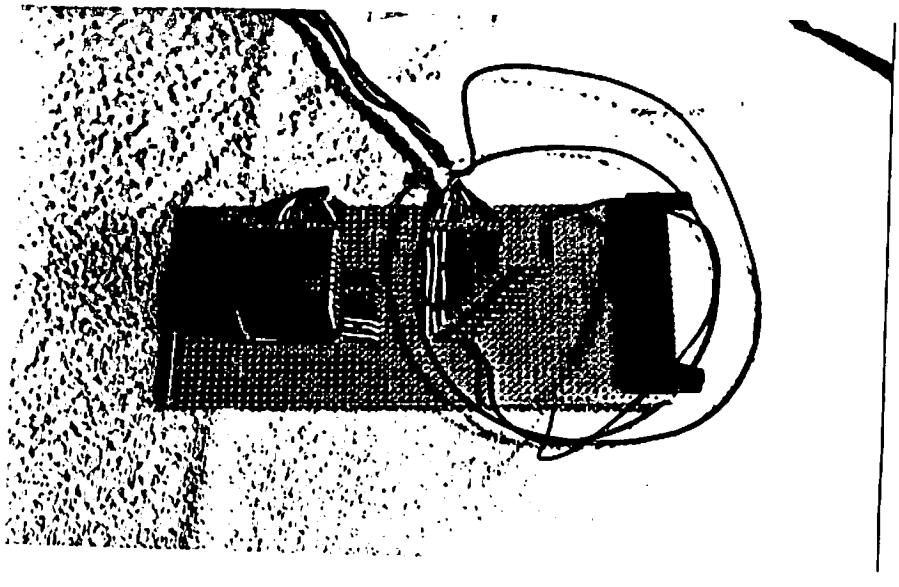


Figure 2.14

Version 1 of the sensor used in this work. The top protective layer has been removed to expose the upper set of conductors. The upper ends of the carbon-filled rubber columns are also visible. This device has a spatial resolution of 100 force elements, forcels, per square inch.

2.4.1 Version 1

The initial design was intended to demonstrate the concept in a bench version. It was not intended that this version be mounted on a gripper. The sensor contains 10 rows of 10 transducers in a one inch by one inch package providing spatial resolution of 100 transducers per square inch. The overall dimensions of the sensor are 1 1/8" x 1 1/8" x roughly 1" in height. The sensor is mounted on a small board which contains the dropping resistors and a connector providing an interface to the scanning computer [157]. This configuration is shown in Figure 2.14.

The basic material for the pad was developed for use in connectors for the electronics industry [165], [192]. The sensor pad contains 100 evenly distributed carbon-doped elastomer columns embedded in a 1"x1"x.2" insulating rubber block. Each column is .035" in diameter. The pad was constructed from ten 1" long 10-conductor dedicated LCD connectors manufactured by the Tecknit Company [197]. These connectors consist of a number of carbon-doped cylinders evenly spaced with their axes aligned parallel to one another mounted on an insulating strip. To produce the pad these connectors were placed back-to-front as shown in the right half of Figure 2.15 such that the spacing between the column centers was .1". The spaces between the rows was filled with GE RTV108 silicone rubber to produce a single, solid block. The surfaces of the block were trimmed and sanded to produce the pad shown on the left in Figure 2.15. These basic materials were chosen due to their robust performance characteristics. These materials can withstand corrosive environments and temperature cyles with no lasting effects [49]. While the experimental conditions employed in this work included only normal environments, it was felt that the performance characteristics of the materials would prove useful in future designs. This process proved to be time-consuming and inaccurate but did produce usable pads.

The diode matrix is located directly under the sensor pad. A small PC board was etched which contained one circular pad for each transducer, carbon-doped column, in the pad. Common 914 signal diodes were located under the board. One diode per pad was mounted with its cathode lead protruding through the pad. The leads were soldered to the pads and trimmed. The contact array is shown in Figure 2.16. The anode leads of the diodes in each row were shorted together and one line per row was brought out providing the x lines for the scanning circuitry. The entire diode matrix was potted in fiberglass resin to provide structural support. The horizontal wires in the Figure provide the y lines and connect to solder pads on opposite sides of the cube.

The y lines to the analog converters are also connected to these pads, as are the dropping diodes. The other side of the dropping diodes are connected to signal common. The pad and fine wires were covered with a layer of fabric tape. This was to stabilize the wires against movement relative to the pad and to provide a degree of protection against damage by external objects.

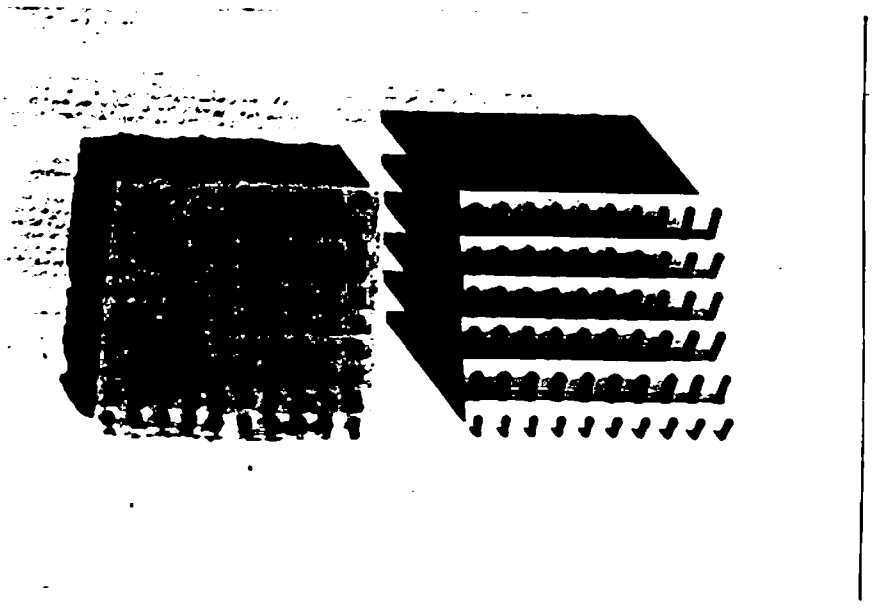


Figure 2.15

This figure shows the construction process used to produce the sensor pad illustrated in Figure 2.11 and visible in the version 1 sensor in Figure 2.14 The pad is constructed from ten 1″ long 10-conductor dedicated LCD connectors consisting of a number of evenly spaced carbon-doped cylinders. The pad is produced by placing these connectors back-to-front as shown in the right half of the figure so that the spacing between the column centers is 0.1″. The resulting spaces between the rows is filled with GE RTV108 silicone rubber to produce a single, solid block. The surfaces of the block are then trimmed and sanded to produce the final pad.

The sensor provided a great deal of experience with the fabrication details and continued to function until it was retired in favor of a later version. Several problems became apparent in this sensor. In order to increase the spatial density of the transducers, the diode matrix needed to be reduced in size. Also, the wires had a tendency to drift and were easily kinked by sharp objects.

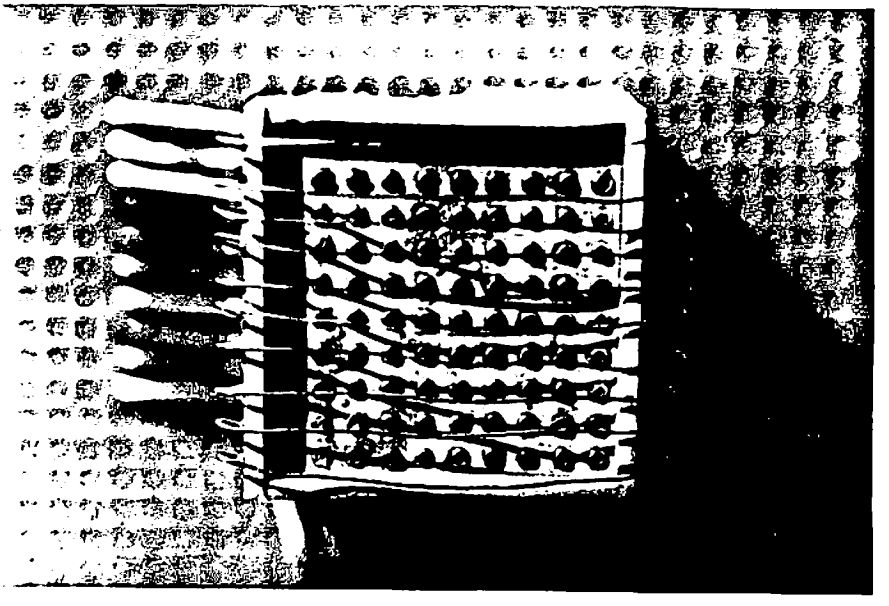


Figure 2.16

This shows the array of contacts in the first version of the sensor. This is the same device as shown in Figure 2.15 except that the sensor pad has been removed.

2.4.2 Version 2a

The next phase consisted of focusing on two things: increasing the spatial density and providing a cleaner test fixture. The resolution was increased from 10x10 to 16x16 per square inch. This change necessitated a change in the pad fabrication technique since the dedicated LCD connectors are readily available in only 10 conductors per linear inch and the total thickness on the conductor/carrier strip is greater than 1/16".

The Tecknit company was contacted in regard to the availability of the .035" diameter conductive rubber material used for the columns in the dedicated connectors as a separate product. The material is available as round cord in arbitrary lengths under their product name SC-Consil [58]. A mold was constructed which holds a square array of 256 lengths of this cord aligned in parallel between the ends of a rectangular box. The box is filled with General Electric RTV-11 silicone rubber. The resulting block was sliced into sheets along planes orthogonal to the cords. This procedure produced pads with 256 transducers per square inch.

The base and printed circuit board were redesigned to include this change in resolution and to be more amenable to experimentation. They appear in Figure 2.17.

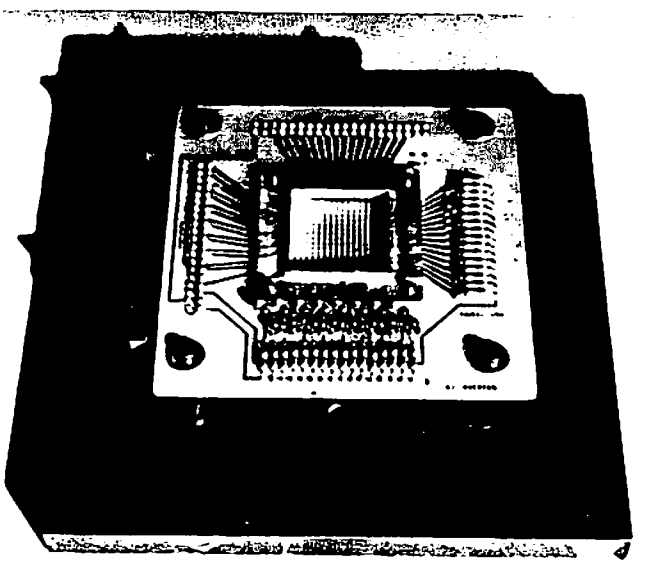


Figure 2.17

This figure shows the circuit board and test stand for the second version of the sensor. This device has a spatial resolution of 256 forcels per square inch arranged in 16 rows of 16 columns. This version was designed as a vehicle for testing different approaches to fabrication of the top set of conductors.

Glass-encased signal diodes were used in version 1 but the N049 case size precluded their use in this version. Instead, chip diodes were employed. The chips measure .015"x.015"x.008" thick. The anode contact is provided by a .0055"x.0055" evaporated aluminum patch centered on one face of the block. The entire opposite face is covered with evaporated gold and provides the cathode contact. The unpackaged diodes were mounted directly onto the PC board. The PC was etched with a small pad for each diode. A drop of electrically conductive epoxy was placed on each pad. With the use of a micromanipulator, a single diode was located, anode side down, on each of the drops of epoxy. Finally, the upper face of each diode was covered with gold ball-bonds to provide a better contact surface.

The transducers must contact only the top, cathode, of the diode chips. The entire PC board, including the edges of the diodes, was coated with a film

of insulating epoxy. The only conduction paths were then from the ends of the etch lines at the edge of the matrix through the diodes and ball-bonds. The area of contact between a diode and a carbon-doped column was quite small, exactly the size of the cathode face of the chip. In order to increase the contact area, the cathode surface was covered with gold ball-bonds. The ball bonds appear as small globs of gold stuck to the evaporated gold surface of the diode. The entire process is diagrammed in Figure 2.18.

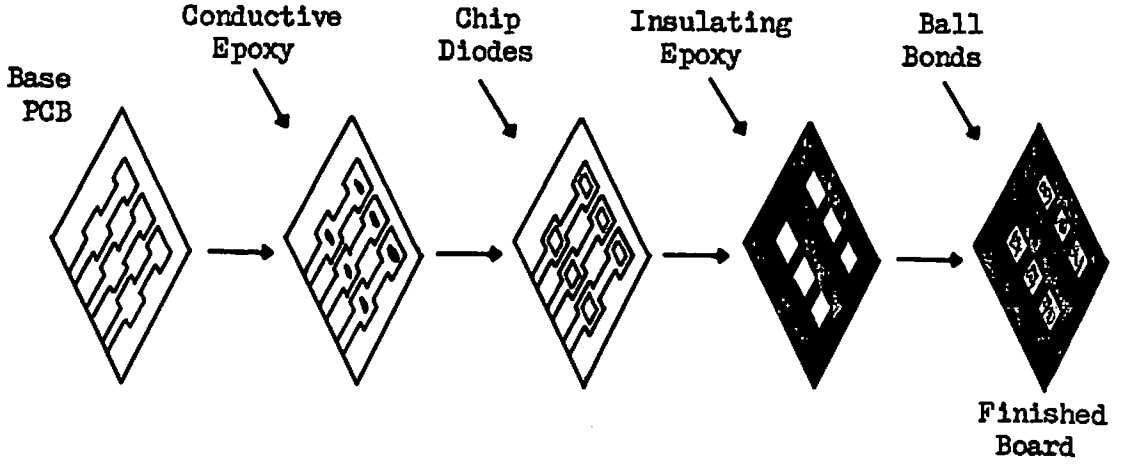


Figure 2.18

This contains a block diagram of the fabrication process for the generic tactile sensor printed circuit board. The process begins with an etched and cleaned PCB. A drop of silver-filled epoxy is placed on each of the pads and then a diode is placed on the epoxy. This assembly is baked to cure the epoxy. Following this, the entire board, with the exception of the upper surface of each diode, is covered with a layer of insulating epoxy and again baked. The final step involves covering the exposed surface of the diodes with a layer of ball-bonds made from 0.001" diameter gold wire. This produces a rough, highly conductive surface.

As in the previous version, the contacts to the top ends of the columns were provided by fine wires. This version proved successful, but two problems remained. The wires exhibited the same problems discussed in regard to version 1, i.e. wandering and damage from pointed stimuli. In addition, no effective means had been developed to affix the pad to the PC board.

2.4.3 Version 2b

The problems associated with the metallic nature of the top conductors were the focus of this version. In this case Version 2a was modified by replacing the fine wires with strips of silver-doped rubber. This material has the properties of low resistance and flexibility, cf. the discussion of the generic sensor design. The rubber strips were cut from material obtained from Tecknit in the form of a .020" thick sheet under the Consil-II product name [59]. The conductive rubber strips were separated and insulated from one another by plain rubber strips of the same dimensions.

While this configuration worked for a short time, the silver-doped rubber strips soon became nonconductive. It was found that pressure applied with pointed objects could change the resistance of the material. Furthermore, prolonged bending of a strip also affected its resistance characteristics. In short, any sharp bend or puncture would cause the strip to lose its conductivity across that point. No real attempt was made to develop an acceptable method for joining the pad and diode matrix in this version.

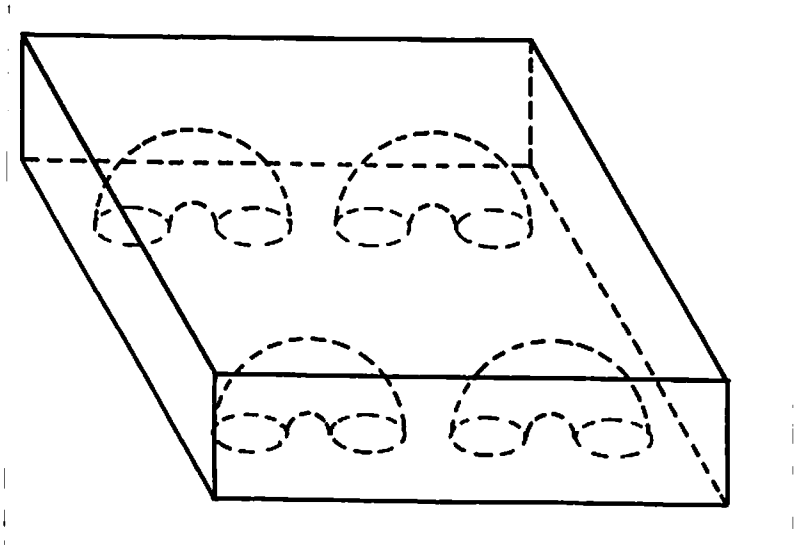


Figure 2.19

The design of the sensor pad for the version 3 sensor is shown in this figure. One problem with the previous designs was that the top set of contacts was vulnerable to breakage due to their proximity to the outer surface of the sensor. In this design, the conductive rubber columns are folded so that both contacts are on the bottom surface of the board. This results in a device with a smooth surface on the side which interacts with the stimulus.

2.4.4 Version 3a

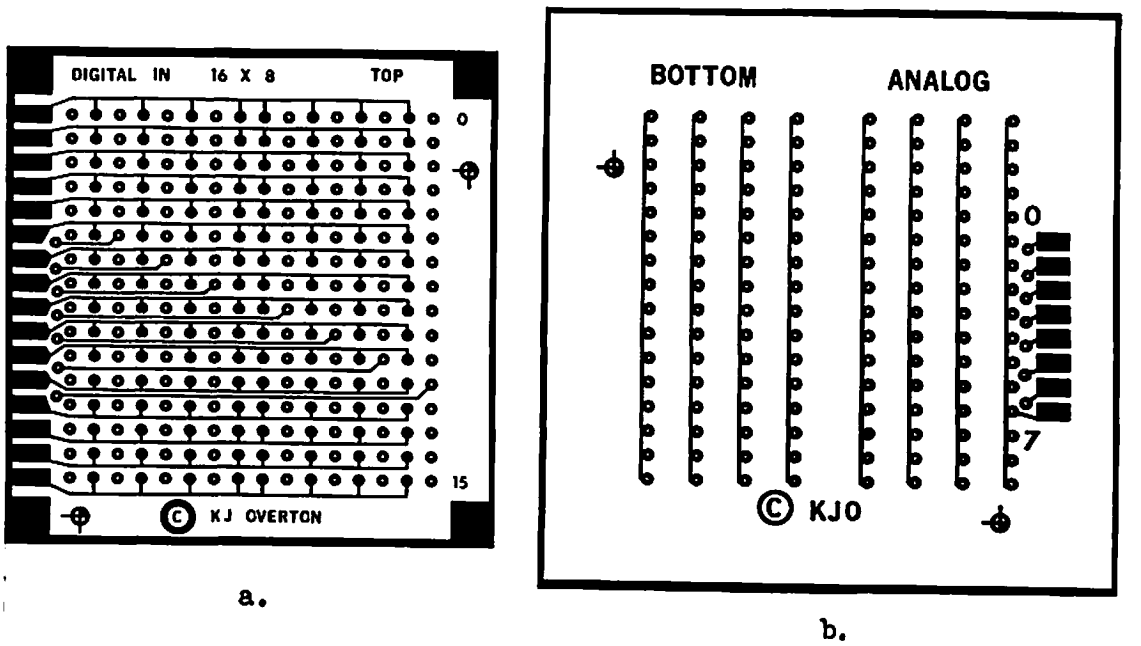
It became apparent that one of the major problems surrounding the previous versions involved the top connections. This third version eliminates the top conductors completely at the expense of decreased spatial density. By altering the shape of the carbon-doped transducers in the sensor pad, both contacts can be placed on the printed circuit board.

In the previous versions the transducers were cylinders with one end on each side of the pad. This dictated that one of the contacts had to be at the interface between the sensor and the environment. In this version the carbon-doped cord is formed into loops. Both ends of the loops are on the same side of the pad. The side of the pad which contacts the PC board then contains all of the contacts while the side of the pad which contacts the environment has no contacts. This design of the sensor pad is shown in Figure 2.19.

A mold was constructed which holds the conductive rubber cord in small, uniformly shaped and accurately positioned loops. The array of loops is then potted in GE RTV-11 silicone rubber to produce the pad.

Moving the top contacts to the same side of the pad necessitated a change in the PC etch pattern. The top, or diode, side of the board is shown in Figure 2.20a. The large rectangular contact pads along the left edge of the board provide contacts for the x lines discussed in the analysis section above. Each solid round pad connected to one of the rectangular pads is a diode mounting site. Each of these pads is paired with an annular pad. These latter pads provide the second contacts for the transducers and are connected to the y lines which are located on the opposite side of the board. The y lines connections are provided by the fingers on the left side of Figure 2.20b. The connections between the etch lines on the bottom of the board and the annular pads on the top of the board are accomplished by passing a 30 gauge tinned copper wire through the board, soldering the wire to the corresponding pads on the top and bottom, and trimming the excess wire. The resulting board has pairs of contacts organized in 16 rows and 8 columns thus providing a spatial density of 128 transducers per square inch.

The diodes used were standard signal diodes and were used unpackaged. These "chip" diodes were glued to the solid pads using a standard, two-part, conductive, heat-cured epoxy. The other contact was provided by a small wire protruding from each annular pad. These wires were soldered in place by hand. Figure 2.21 contains a low-magnification electron micrograph of the sensor PCB after the diodes have been mounted and the wires soldered into the holes. There are eight columns of pairs of contacts. Each pair consists of a diode and a wire/solder bump. The left-right organization of the pairs changes from one side of the board to the other in order to keep the diodes away from the edge of the board.



a.

b.

Figure 2.20

PCB etch patterns for the version 3 sensor. 2.20a shows the top, or diode/sensor pad, side of the board. The large rectangular contact pads along the left edge of the board provide contacts for the x lines. Each solid round pad connected to one of the rectangular pads is a diode mounting site. Each of these pads is paired with an annular pad which provides the second contact for the transducers and is connected to the y line located on the bottom of the board. Figure 2.20b shows the bottom of the board. The y lines contacts are provided by the fingers along the right side. The bottom etch pattern is oriented such that these fingers are aligned with the fingers on the top of the board. The connections through the board are made by feeding 30 gauge tinned copper wire through the holes in the board and soldering them to both sides.

Figure 2.22 contains a micrograph of the board at higher resolution. The diodes are easily discernible as small squares, cf. item A. The epoxy bonding the diode to the pad appears as the white puddle in which the diode rests, cf. item B. The diodes are connected to the x lines in the scanning system by the

Figure 2.21

The pictures in both this figure and Figure 2.22 are electron photomicrographs of the diode array used for the version 3 sensors. This image was taken at roughly 5x magnification. The pads along the left edge of the board provide points of attachment for the wires which connect the transducer to the interface circuitry.

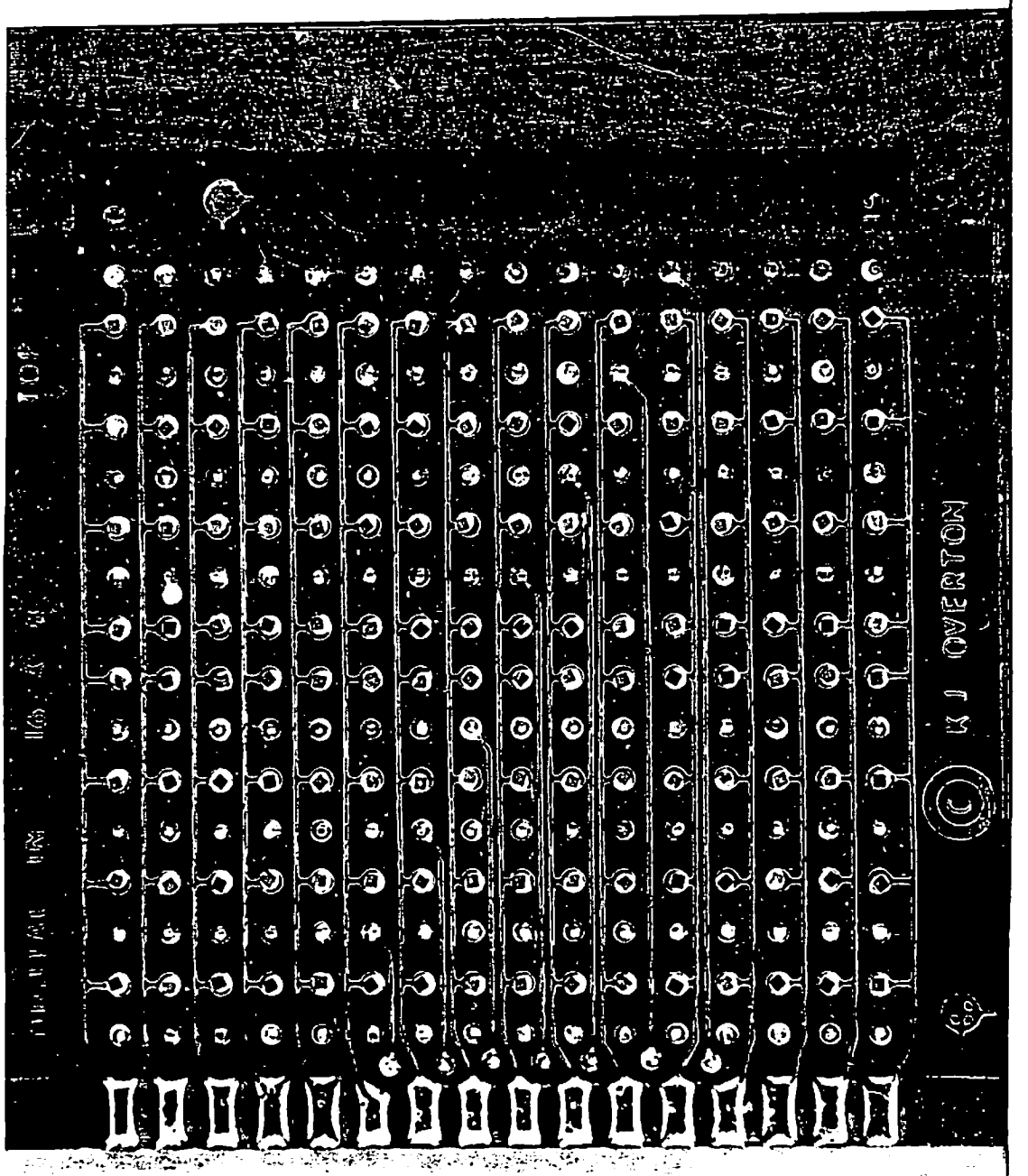
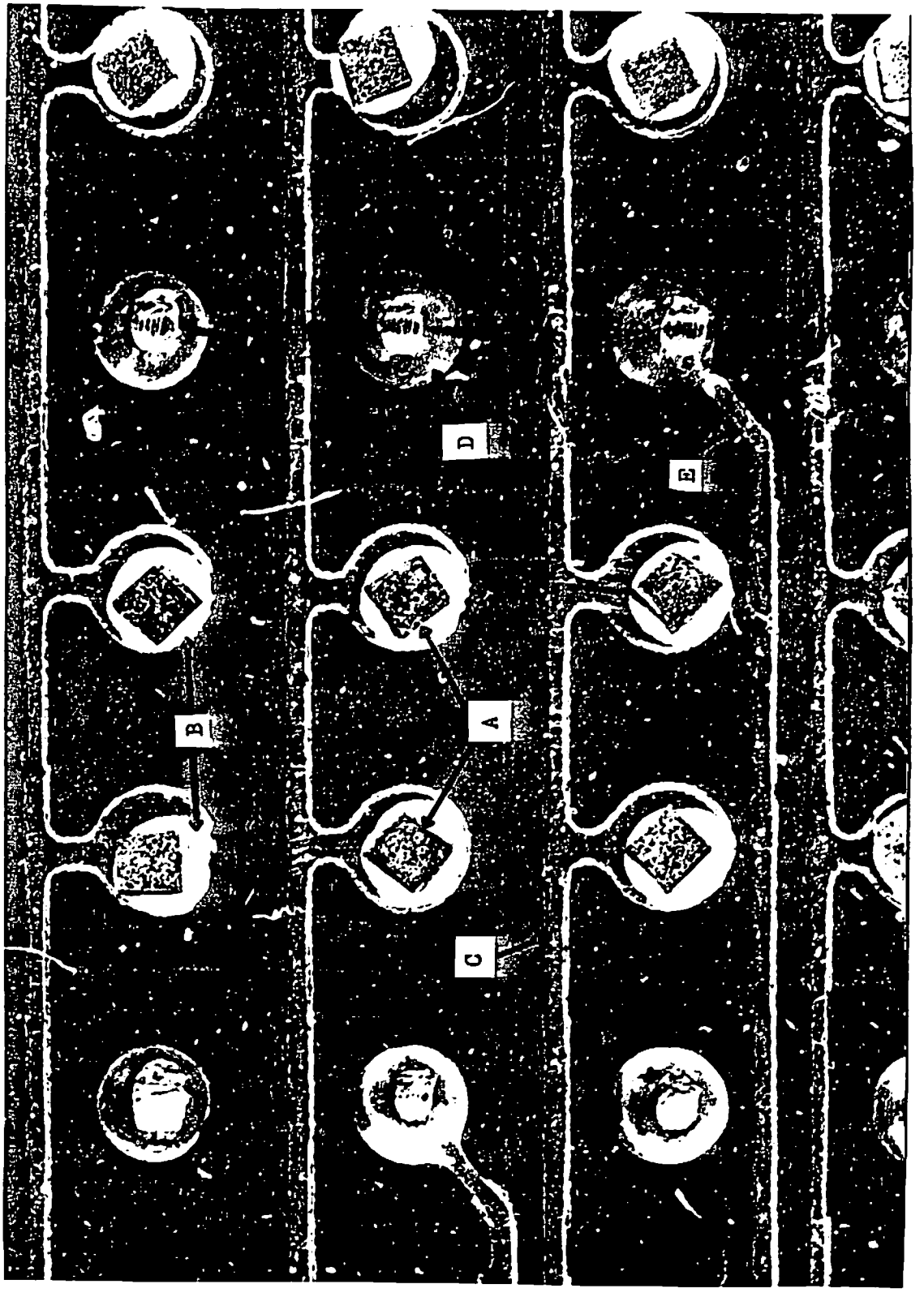

KJ OVERTON

Figure 2.22

This is an electron micrograph of the diode array at 25x magnification. The chip diodes are the rectangular blocks labeled "A". The drops of silver-filled epoxy attaching the diodes to the board are labeled "B". The x lines are indicated by the label "C" while the wires soldered to the y lines on the bottom of the board are labeled "D". The etch lines labeled "D" are jumpers used to route the y lines on the bottom of the board to the contact pads along the edge of the board.


B
A
C
D
B

etch traces on the PCB marked by the letter C. The solder joints and trimmed wires are visible in the Figure and are indicated by the label D. The remaining horizontal traces in the PCB, labeled E, are used to help route the contacts from the wire/solder pads to the fingers on the other side of the board.

Once this phase of construction has been completed, the etch traces on the board must be coated so that the only conduction paths occur through the diode and wire contacts. This is accomplished by carefully flooding the board with an insulating epoxy leaving only the tops of the diodes and the pads with the wires clear. The top of each diode is then covered with gold ball-bonds created from 0.001 inch diameter gold wire with a die bonding machine. These serve the purpose of providing a rough contact surface and the use of gold results in a smaller buildup of corrosion which would impede conduction. Figure 2.23 contains a photo of the completed board. The 7-strand, 28 gauge wires connecting the fingers on the board to the scanning circuitry are soldered to the board making the board ready for final assembly.

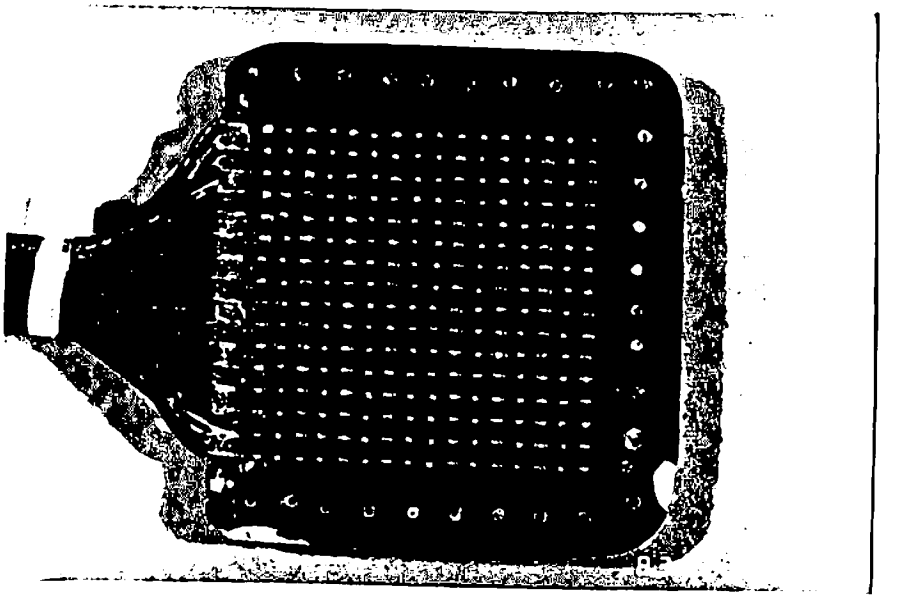


Figure 2.23

This is the sensor PCB ready for assembly. The square array of contacts is clearly visible as is the bundle of wires which connect the transducer to the interface circuitry. The board has been shaped so as to eliminate sharp corners and the set of holes for the pad attachment washer have been drilled.

Sensors built with this design provided a great deal of service and formed one of the basic experimental tools used for the work discussed in the succeeding chapters. While problems with the top connections were effectively eliminated in this version, the question of pad attachment remained.

2.4.5 Version 3b

The question of attachment was addressed in this final version. Here the pad is molded so that the top surface has a raised center section containing the transducer array surrounded by a lower section as shown in Figure 2.24. The PC board is trimmed such that it retains a border section around the matrix. The pad is located on the PC so that the ends of the transducer loops are

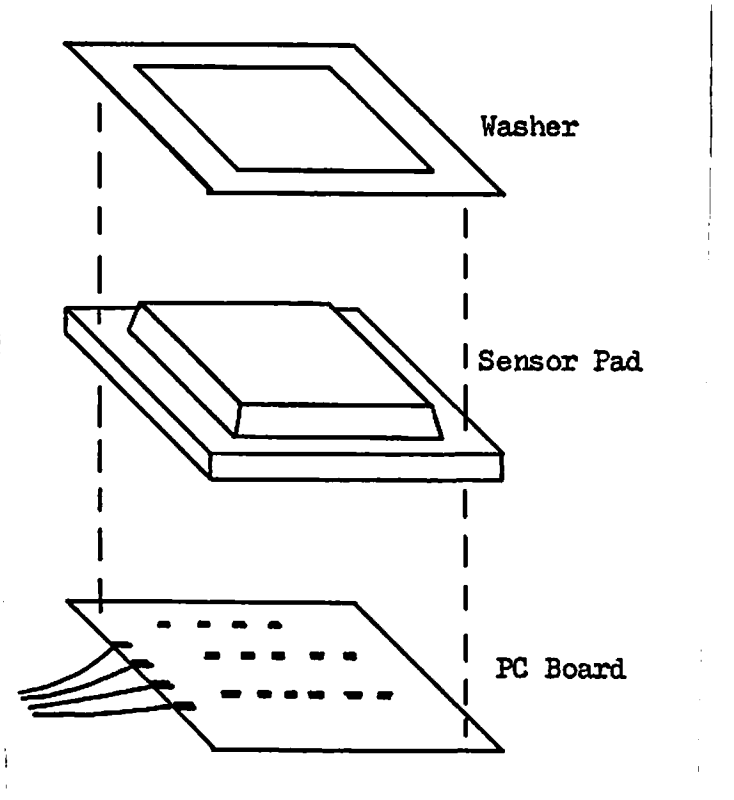


Figure 2.24

This outlines the assembly process for the version 3 sensor. The components are the finished PCB as shown in Figure 2.23, the molded sensor pad, and a thin metal washer. The pad houses the transducers in a raised center section surrounded by a lowered area. The washer fits over the center section and rests on the lower rim. The assembly is held together by fine monofilament stitches.

aligned with the contacts and held in place by a thin, square metal washer. Fine monofilament lines actually sewn through the washer-pad-board stack rigidly hold the pad in position on the board. Figure 2.24 shows the three components of the sensor prior to final assembly.

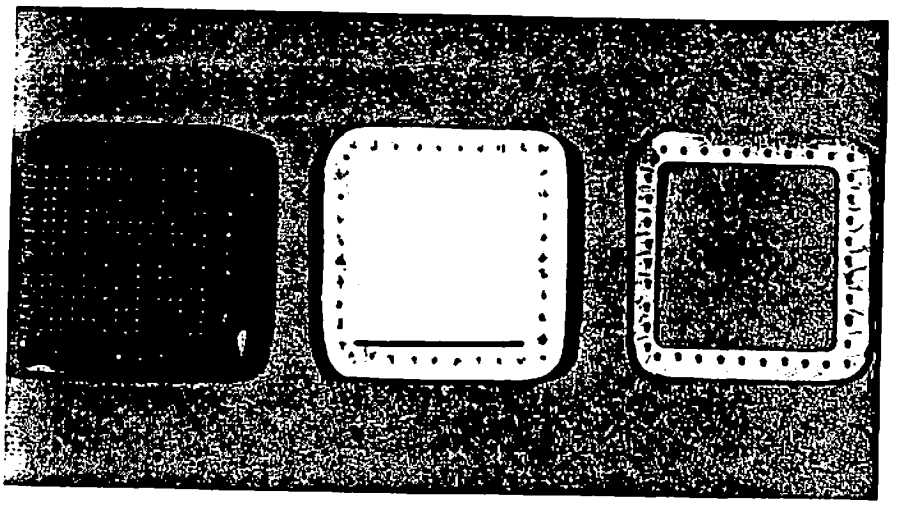


Figure 2.25

This photograph shows the three components as described in the caption of Figure 2.24 prior to assembly.

Version 3b provides reliable, rugged sensors with 128 forcels per square inch in a package which can be mounted on the fingertips of a robot gripper. The forcels are rectangular in cross section with an aspect ratio of 2 to 1 and are arranged in 16 rows of 8 columns. A "correction" process is used to derive a 16x16 force image from the raw 16x8 image. The response characteristics and correction methods form the substance of Chapter 3. The final figure in this chapter, 2.26, shows one of the finished sensors which was used for the experiments in the succeeding chapters.

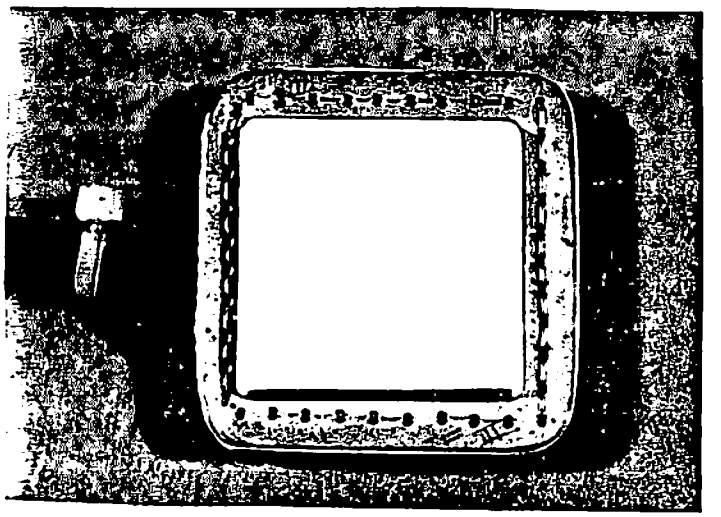


Figure 2.26

A finished version 3 sensor containing 128 transducers in a 1"x1" area. The overall package size is approximately 1.4"x1.4"x0.3" thick.

Chapter 3

Response Characteristics

3.1 Introduction

The topics discussed in the previous chapters include robot senses, robot sensors, and tactile sensors in particular. Chapter 2 focused on the tactile array sensor developed and used during this work. The resulting sensor, shown in Figure 2.26, is a device which provides an array of numbers corresponding to some event occurring at the sensor surface. The question that now presents itself concerns the response characteristics of the sensor. Beyond this, questions of how the tactile data are used must be considered. These questions include the means by which tactile images are "corrected" as well as the algorithms used to extract useful information for the images.

The array of numbers obtained by scanning the sensor contains a great deal of information about the shape of the object sensed, as deflections, as well as the distribution of forces present at the interface with the object. Since the transducer is a physical device, it exhibits non-ideal performance. This includes dead, or non-responding elements, element sensitivity and dynamic range (which vary over the device), hysteresis, etc. Due to the nature of the sensor and construction process, the individual transducers vary in response from one another. Unfortunately, this means that the number produced by one transducer for a given force or deflection may not be the same number produced by another transducer under the same conditions. In order for the spatial information inherent in the nature of the array sensor to be utilized, the relationship between the applied force or deflection and output must be known for each forcel. Once known, a correction technique may be applied which normalizes the output of each transducer to a common range. The computational overhead of the correction algorithm as well as its effectiveness are of interest since we not only need to correct images but to do so in an amount of time which is small enough to allow the transformed image to be processed and the resulting information used in control.

This chapter is devoted to a discussion of the response characteristics of the sensor developed in Chapter 2. The chapter begins with a discussion of the form of the response, including that of the scanning computer, the hysteresis, and continues through a discussion of various algorithms designed to correct tactile images. Chapters 4 and 5 are devoted to experiments in the area of tactile image processing.

3.2 Response Characteristics

This section discusses the response characteristics of the sensor test bed consisting of a DEC MINC and a tactile array sensor. As mentioned in Chapter 2, the ideal tactile sensor would exhibit an number of desirable characteristics, all of which would fall into two general classes. The first class contains those characteristics pertaining to the physical structure of the device, e.g. physical

ruggedness, reasonable fabrication cost, and high spatial resolution. The second class involves those characteristics relating to the response of the device, e.g. low hysteresis, high sensitivity, large dynamic range, the time course of the response, and the monotonicity of response. This section discusses several of the issues in the second class. In particular, the three areas which are considered below involve the time course of the response, the form of the response, i.e. its monotonicity, and the hysteresis. Issues of the force sensitivity, dynamic range, and spatial resolution are discussed in a subsequent section.

The DEC MINC, or Modular Instrumentation Computer, uses an 11/23 plus processor, an analog-to-digital converter module, and a digital output module. The sensor is composed of two logical parts: the transducer itself and the circuitry which connects the sensor to the computer. The latter consists of the dropping resistors, lead wires running from the sensor proper to the computer and the connectors. The diode array and printed circuit boards are considered part of the sensor transducer since these components and the rubber pad form a single unit. There are several aspects to the overall response characteristics of the system.

The response of the scanning circuitry is essentially defined by the time required to scan a selected transducer, the time required for the input signal to settle, the time required to convert the signal to digital form, and the linearity of the analog to digital conversion. One of the first points to consider is the time required to obtain the value for a given forcel in the array. This value will determine the maximum scanning frequency for the device. Also of interest are the linearity of the response and hysteresis of the transducers. Since the sensor pad is a physical substance and must contact the object being sensed, there will be some spreading of the response of the transducers to sharp stimuli, i.e. there is a points spread function for the sensor. Knowing the magnitude and form of this function will be useful during processing of the force images. Each of the above characteristics should be determined for the two parts of the test bed. In some instances, no differentiation will be made as to the particular characteristics of the components of the sensor. Lumping the characteristics of the components of the sensor does not cause any problems during the image processing phase.

3.2.1 Scanning Hardware

The first system to consider is the scanning computer since the characteristics of the devices are known and the values for any unknown parameters are easily determined. The process of scanning the sensor and collecting forcel data relies on two parts: software and hardware.

The hardware consists of an 11/23 plus processor, A/D hardware, and a digital output board. Recall that the first phase of data collection requires that the lead wire corresponding to the row containing the forcel of interest must be driven high while the remaining lines are held low. This process is accomplished in software and occurs electrically at the outputs of the digital output board. In the form used during all of the experimentation in this thesis, the software and hardware combined required approximately 100 microseconds to select a

given digital channel, pull it high, and take the remaining channels low. The typical instruction execution time for the 11/23 processor is on the order of 2.0 microseconds. Once the appropriate digital line has been chosen, the analog channel corresponding to the column address of the desired forcel is selected and the analog voltage on the line converted to a digital representation. The total time required for this phase is approximately 170 microseconds. This figure represents the sum of the acquisition/conversion time of the hardware A/D and the execution time of the controlling software.

The A/D converter has 12 bits of resolution over an input signal range of plus and minus 5.12 volts with plus and minus one bit linearity in its single-ended mode. The A/D has a 200 nanosecond aperture time for the sample and hold circuitry and a typical conversion time of 40 microseconds [134]. The controlling software requires approximately 135 microseconds per analog channel resulting in a total data acquisition time of roughly 175 microseconds. Given these figures for the digital selection time, td, and the analog data acquisition time, ta, the minimum total time required to scan a tactile array sensor with "n" digital channels and "m" analog channels is calculated from:

$$T = (n \times td) + (n \times m \times ta)$$

For the array sensor of interest here, t is approximately 24 milliseconds. This translates to a scanning frequency of slightly over 41 Hz. These figures are for an ideal system with no overhead other than the scanning code and ideal response from the analog hardware.

One question which arises is how quickly the analog to digital converter settles to the input voltage. The specifications of the device provide a figure of 40 microseconds for the conversion time. What is of interest here is not this figure, but instead, the time that the analog inputs take to settle to the actual input voltage. A simple experiment was performed to understand this.

In this experiment a regulated power supply and resistor ladder were used to apply stable voltages directly to the input channels of the A/D device. Several voltage arrangements were used including: positive voltage step between adjacent channels, negative voltage step between adjacent channels, and positive and negative five-volt steps between adjacent channels. Figure 3.1 contains the graphical results of this experiment. Only four channels are shown in order to keep the amount of data to a manageable size.

The eight input channels to which the array sensor is connected were tested and it was found that the output of the A/D device does not vary from the first to subsequent readings. This implies that the above scanning time calculations are valid, given a sensor circuit which can respond fast enough.

The next question, then, involves the speed of response of the sensor. Figure 3.2 shows the response of four forcels over time. In this case the readings are approximately 175 microseconds apart. One sees that in some cases a sensor's response is nearly flat over the entire set of readings. (It should be noted that there is an inaccuracy of 1/2 bit in the hardware as noted above, but this

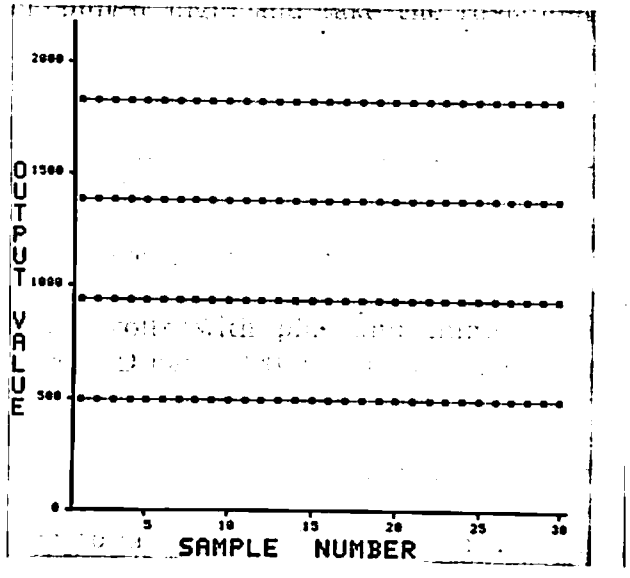


Figure 3.1

A regulated power supply and resistor ladder were used to apply stable voltages directly to the input channels of the A/D device. Several voltage arrangements were used including: positive voltage step between adjacent channels, negative voltage step between adjacent channels, and positive and negative five-volt steps between adjacent channels. The results were identical in all cases. Each square represents one reading from the analog channel. Only four channels are shown in order to keep the amount of data manageable.

variation is far too small to be noticed here. The slow undulations are due to changes in the stimulus.) In other cases the response seems to start either below or above the final value and move toward the steady state as time progresses. The variation is generally restricted to 20 units regardless of the actual input value, i.e. the maximum deviation is plus or minus .1% of the full scale reading. There is almost no difference seen between the second reading and the remainder of the trace.

There are clearly two extremes in response form. In one case the output starts at a value below the final value and increases during the first few readings until it reaches the final state. Alternately, the output may begin above the final value. The question here relates to the cause of the non-steady response. One possibility is that this phenomenon is due to the time constant of the sensor circuit. Recall that the sensor is basically a variable resistor. The circuitry on the PCB holding the diode matrix and the long lead wires have certain

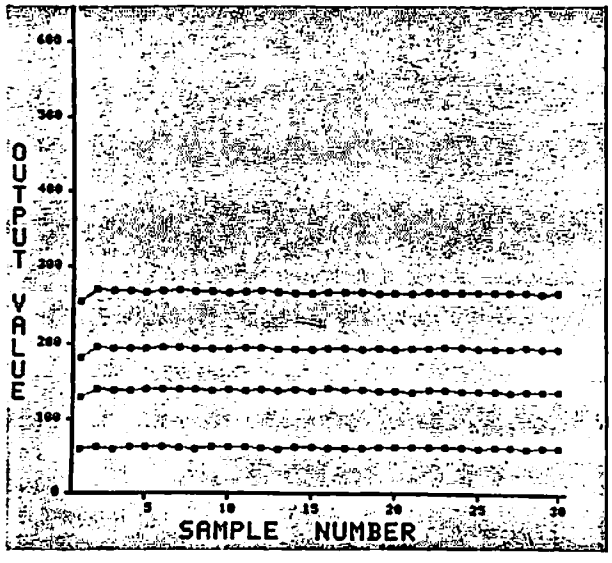


Figure 3.2

This shows the response of four forcels over time. The readings, indicated by the squares in the graphs, are approximately 175 microseconds apart. In some cases the forcel's response is nearly flat over the entire set of readings. In other cases the first reading varies from the rest.

capacitances. These capacitances combined with the resistance inherent in the sensor produce a time constant which may be long enough to become a factor during scanning. A typical value for the resistance of a sensor element is 50K ohms. Considering this value and a value of 200 microseconds for the time constant yields a capacitance of 40,000 picofarads. This value seems inordinately large considering the type of circuit.

An alternate possibility is that time is required to back-bias the diodes when a particular digital line is selected. The settling time seen experimentally seems to be very large relative to what the time to bias the diodes should be.

If the amount of time needed for the digital value to settle is dependent upon either an RC time constant or the time required to bias the diodes when a digital channel is chosen, then once a digital channel has settled the readings from the remaining channels should not vary. However, the increasing/decreasing behavior is typically observed at almost all locations in the array. In particular, it is seen for many columns, or analog channels, on a given row, or digital channel. Furthermore, the amount of variation changes from one transducer to another and from one test to another for a given transducer. If this rise time is due to a time constant inherent in the sensor, then the insertion of a delay greater than the time required for one A/D conversion should allow the input

to the A/D to stabilize. A 6 millisecond delay between the selection of a digital channel and the conversion of the first analog channel, and between successive analog channels, resulted in the behavior shown in Figure 3.3. As is obvious from the figure, the transducers still respond with the first value usually above or below the subsequent readings.

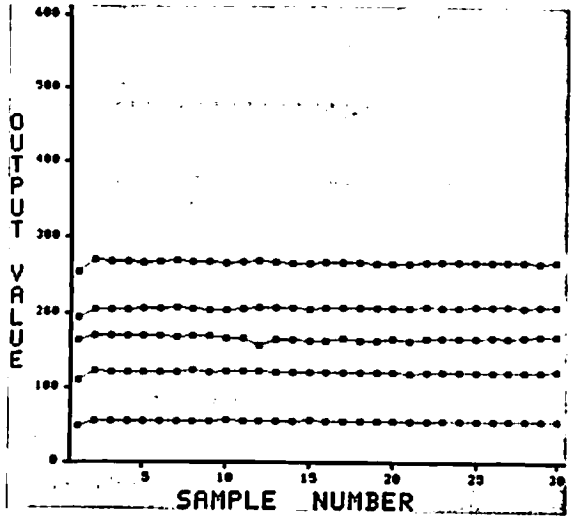


Figure 3.3

This graph was generated in the same manner as that in Figure 3.2 except that a 6 millisecond delay was added between the selection of a digital channel and the conversion of the first analog channel and between successive analog channels. The transducers still respond with the first value usually above or below the subsequent readings.

The obvious next test involves inserting a delay only between the selection of the analog channel and the initiation of the first A/D conversion. A software delay of roughly 1.5 milliseconds was used and the results are given in Figure 3.4 As in the previous tests, the applied force was maintained at a constant value during scanning. Several force values were used in different tests with the same results. The figure shows that the transducers no longer exhibit a variation in their first response.

From this it is safe to assume that this time behavior is due to interactions between the sensor circuitry and the analog multiplexer and sample and hold circuitry in the MINC system. It is also clear from the figures that the amount of time required for a given channel to become steady is rarely more than one conversion time. This allows the formulation of a heuristic to be used in the collection of force images. The heuristic states that the datum for a given channel is valid beginning with the second analog to digital conversion.

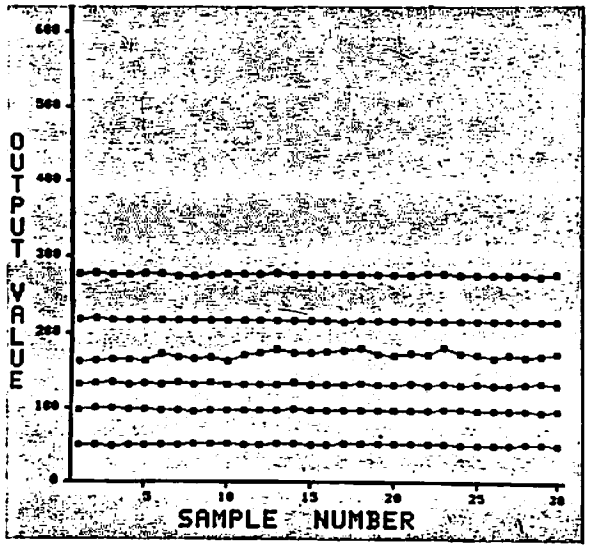


Figure 3.4

In this case a 1.5 millisecond delay was inserted between the selection of the analog channel and the initiation of the A/D conversion. In this case the variation in the first reading from a channel is not noticeable.

The previous discussion focused on the response characteristics of the hardware used to interface to the circuit. The remainder of this section concentrates on the sensor itself. As discussed earlier, there are several response characteristics which need to be understood in order to fully utilize the sensor. The first aspects to consider are the linearity of the response and repeatability of the transducers. This includes understanding not only the form of the response of a given transducer but also the variation between transducers. Closely related to the linearity are the sensitivity and dynamic range of each transducer. Sensitivity refers to the smallest force or deflection detectable by the device. A consideration entering into this is the noise present in the data obtained from the sensor. The hysteresis characteristics are of interest since they must be considered during both static and, more importantly, dynamic tactile image processing. The final point to consider involves how the sensor responds to a point stimulus, i.e. what is the spread function of the device. This information and Nyquist's principles provide insight into the theoretical limits of spatial resolution. A series of experiments designed to address these issues was performed on three sensors and the results are presented below.

3.2.2 Structure of Testing System

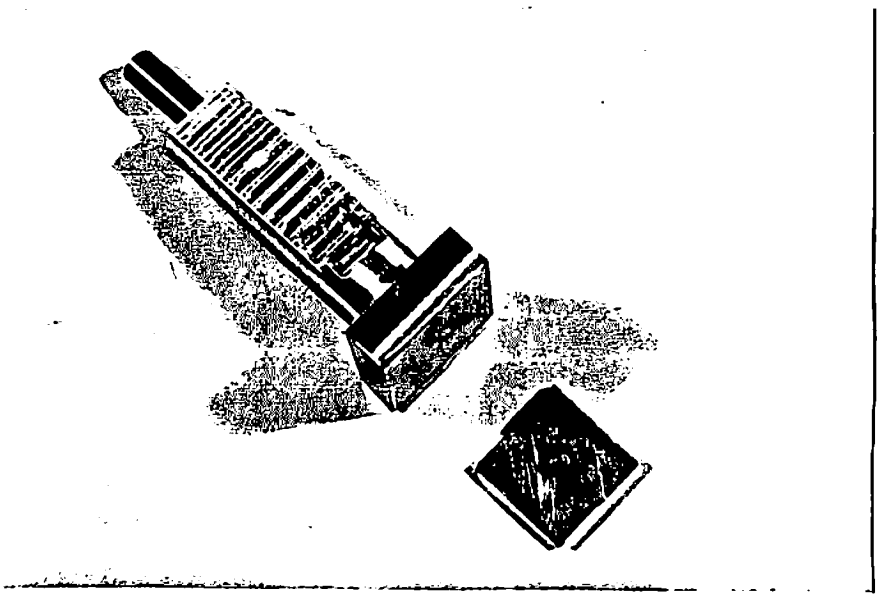
In each case where the sensor was probed, an IBM RS1 manipulator system was employed. The sensor was scanned by the DEC computer running FORTRAN and MACRO software as described in previous sections. A control program written in AML running on the IBM Series 1 (control computer for the RS1 manipulator) was used to conduct the experiment. The AML program moved the probe, or sensor, and then requested tactile processing to be done by the DEC computer. Communication between the IBM Series 1 computer and the DEC MINC was implemented as a 16-bit parallel path utilizing the DI/DO ports on the IBM side and a DRV11 on the DEC side. General purpose data transfer routines were written in AML and MACRO for the RS1 and DEC sides respectively. The details of the interface, both hardware and software are provided in Appendix A.

The tests run here consist of applying either a uniform force over the entire surface of the sensor or applying a force to an individual forcel. In the first case, two methods were employed: in one the sensor was attached to the end of an aluminum "proboscis" which was held between the fingertips of the RS1 and in the other the sensor was placed on the worktable of the manipulator and the pressure applied by a small plate. In the case where individual forcels were probed, special tips were held by the RS1 manipulator.

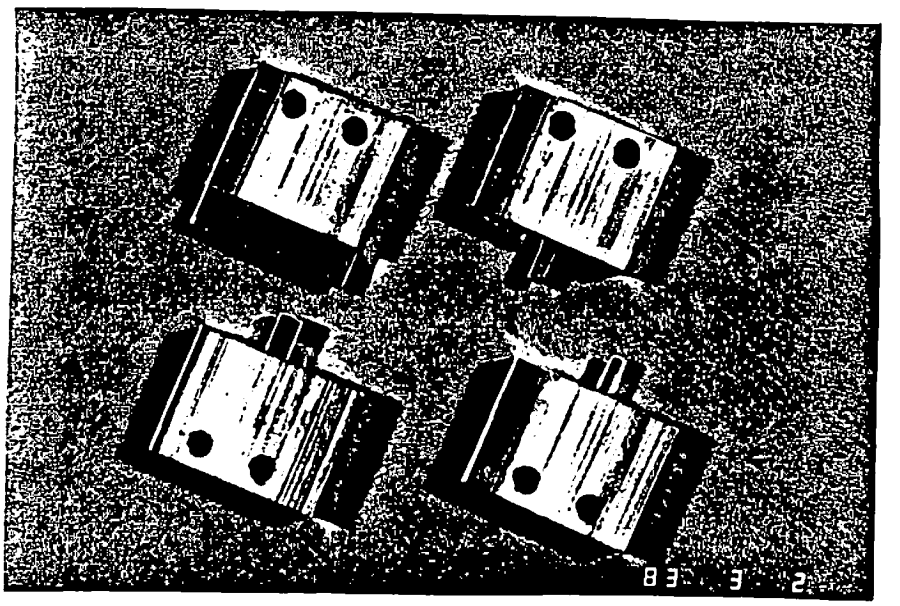
The proboscis is shown in Figure 3.5a and consists of an aluminum shaft with a rectangular end plate to support the sensor. The shaft has parallel flats to allow proper seating of the fingertips. Double-sided tape was used to attach the sensor to the proboscis. The difficulty with this method lies in the fact that the upper and lower surfaces of the sensor are not necessarily parallel. Before the trials were conducted, the surface of the sensor pad was visually aligned with the work surface of the robot. Misalignment tended to cause unwanted stresses in the sensor and thus confounded the calibration information. In addition, there are distinct neighborhood effects using this technique. Each transducer has a particular force value below which it will yield a constant output value. In some cases, force applied to a group of forcels caused a given forcel to begin to respond before the normal force threshold was reached. This behavior is a result of mechanical interactions between the transducers.

Figure 3.5b shows the special probe tips used for the cases in which a single forcel was probed. Three of the probes used have the same aspect ratio as a forcel, i.e. 2x1. The areas of the probes are: 1/2 the area of a forcel, the same area as a forcel, and twice the area. The fourth probe is a long thin stimulus similar to a knife blade. Figure 3.5c shows the holder that was used to facilitate manipulation of the tips by the RS1.

The basic experimental paradigm consists of recording the response of the sensor transducers, either singly or collectively as an image, and the applied force and height of the probe as the surface is stimulated by the manipulator.



a.



b.

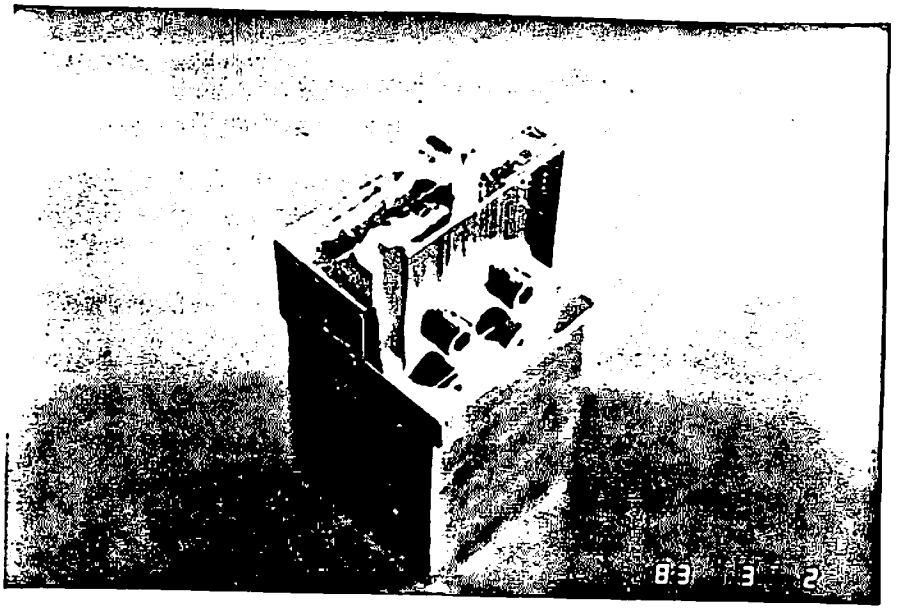


Figure 3.5

Probes used for the response characteristics data collection experiments. 3.5a: The proboscis probe used to hold the sensor for uniform force experiments and later for tactile exploration experiments and the ball and plate assembly used to even the force out, see text. 3.5b: tips used to stimulate individual forcels. One tip consists of a long thin edge while the other three have rectangular cross sections with the same aspect ration as a forcel. The areas are 1/2x, 1x, and 2x the area of a forcel.

3.2.3 Force versus Deflection

Throughout the previous discussions, the terms "force" and "deflection" have been used interchangeably to name the free variable in the calibration experiments. This is due to the fact that the sensor pad may be viewed as a homogeneous, elastic block whose force/deflection characteristics are defined by a spring constant. That is, a given force applied over a given area will result in a predictable amount of deformation of the surface. Similarly, deflecting a given area of the sensor pad a specified amount will require a predictable amount of force. The preceding discussion assumes that the sensor pad, and in particular the force and deflection transduction system, are ideal in nature.

The sensor pad may be modeled as an ideal material due to its homogeneous nature. The response of the testing system, on the other hand, does not respond in an ideal fashion. The force transducers used in the gripper of the IBM RS1 manipulator are resistive film strain gauges. An undetected problem in the strain gauge interface electronics on the IBM RS1 robot used for the calibration experiments resulted in large random fluctuations in output signals. Several months after the experiments were performed it was learned that the problem was the result of moisture trapped inside the IC's containing the strain gauge amplifiers. One unit in the digital output from the force transducers is 1 gram. During the course of experimentation utilizing the RS1, the base-level, zero-force readings from the strain gauges varied as much as 100 counts. In addition, the repeated application of a given force, as determined through interrogating the strain gauges, resulted in widely varying actual applied force.

The position measurement system is based on an acoustic transducer and is accurate to at least four decimal places. The positioning accuracy of the manipulator is .008 inches due to the mechanical structure and the drive system. AML contains commands which interrogate the position encoders as well as the last position to which the manipulator was commanded to move. Thus while the movement of the manipulator is not extremely accurate, the actual position of each of the joints is available.

In the following discussions, and in fact throughout the text, when the data is said to be uncalibrated, the actual output values from the sensor is used. When the data are said to be calibrated, the calibrated values are based on the sensor output versus measured deflection curves.

3.2.4 Format of Data Display

The results from the experiments are presented as graphs showing the sensor output as the dependent variable and either the force or deflection as the independent variable. The abscissa of each graph is either the applied force or the deflection of the sensor pad. The ordinate is always the output of the transducer. The force/deflection, sensor output pairs are indicated by either a square or a triangle with adjacent symbols connected by line segments. The squares indicate recordings during the increasing force/deflection phase of the probing while the points indicated by triangles were recorded as the probe was

withdrawn from the sensor. Thus the difference between the triangular and square symbol traces is an indication of hysteresis in the sensor. In the uniform force experiment with the sensor mounted on the proboscis, the sensor output was collected only as the probe was lowered in the sensor, i.e. hysteresis information is not available in this case. Figure 3.6 contains the graphs for eight transducers from the three test sensors. Figure 3.6a contains a plot of the output versus the applied force and Figure 3.6b contains the data from the same forcels except plotted as output against the deflection.

3.2.5 Experimental Paradigms

Several experiments are presented which concern the response characteristics of the sensor. The experiments are divided into four basic classes: uniform force application, single forcel probing, settling testing, and point-spread testing. Several variations were performed and are discussed in detail below. A discussion of the processing techniques used to calibrate the sensors is given in the next major section.

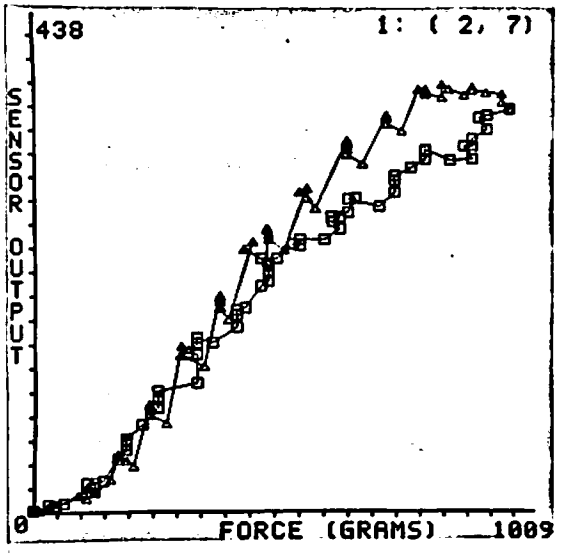
In all classes except "settling", a probe was lowered onto the sensor and the applied force and sensor response measured. The probe was lowered such that the force increased in steps of a given size. Once a desired force level had been attained, the actual force and deflection values were recorded and transmitted to the DEC computer to be saved in a file along with the readings from the sensor. The force was stepped from 0 to a maximum value, depending upon which probe was used, and then stepped back to zero. These traces not only provide linearity information but hysteresis information as well.

3.2.6 Response Form

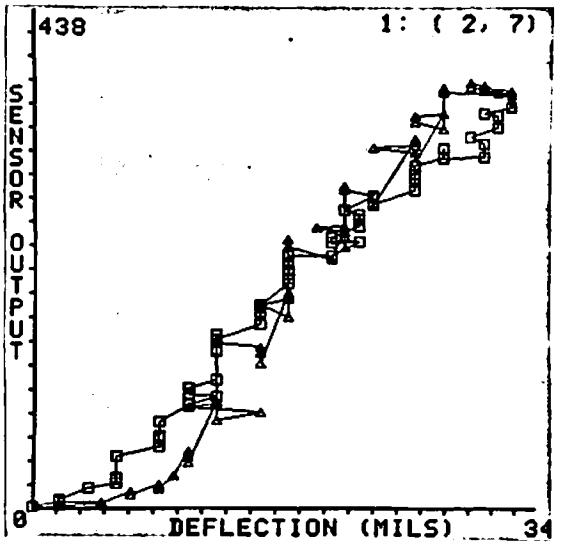
The purpose of this experiment was to investigate the form of the response of the individual transducers. An ideal sensor would exhibit linear output, given a linear stimulus, and would begin responding with an infinitesimal change from zero in the stimulus. That is, there would be no dead zone in the response. Of course, such transducers do not exist; the best one can hope for is that the response is monotonic over the range and that the stimulus value at which the transducer begins to respond is small enough to be useful, as defined by the application domain.

In this case, a small probe was lowered onto the pad by the IBM RS1 manipulator. Three probes were used, each with a cross sectional aspect ration identical to a forcel, i.e. 2 to 1. The areas of the probes were: one half the area of a forcel, the same as the area of a forcel, and twice the area. The sensor was affixed to the workspace of the robot and each forcel probed by the RS1.

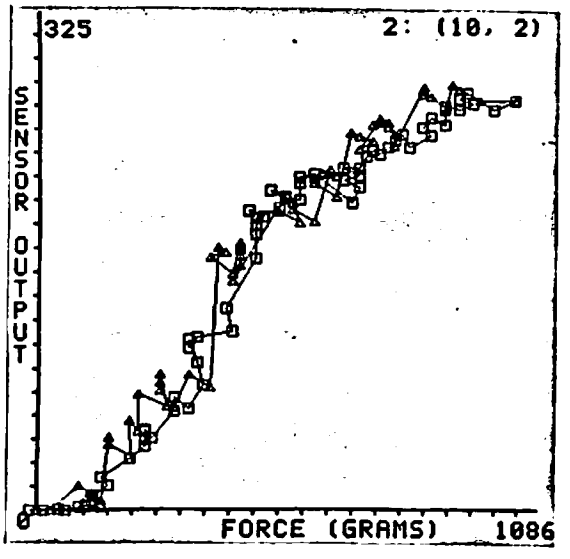
Probing of a forcel consisted of centering the probe over the element and moving the probe into the pad until a predetermined force was obtained. Small force increments were used to direct the probe. The probe was lowered/raised until a desired force level was obtained. After the sensor data were collected, the desired force level was increased/decreased and the probe moved. Once the


438
1: (2, 7)
SENSOR OUTPUT
0
FORCE (GRAMS)
1009

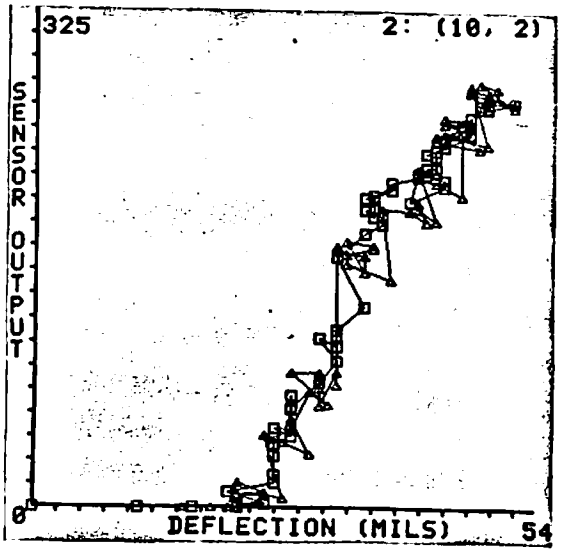
a.


438
1: (2, 7)
SENSOR OUTPUT
0
DEFLECTION (MILS)
34

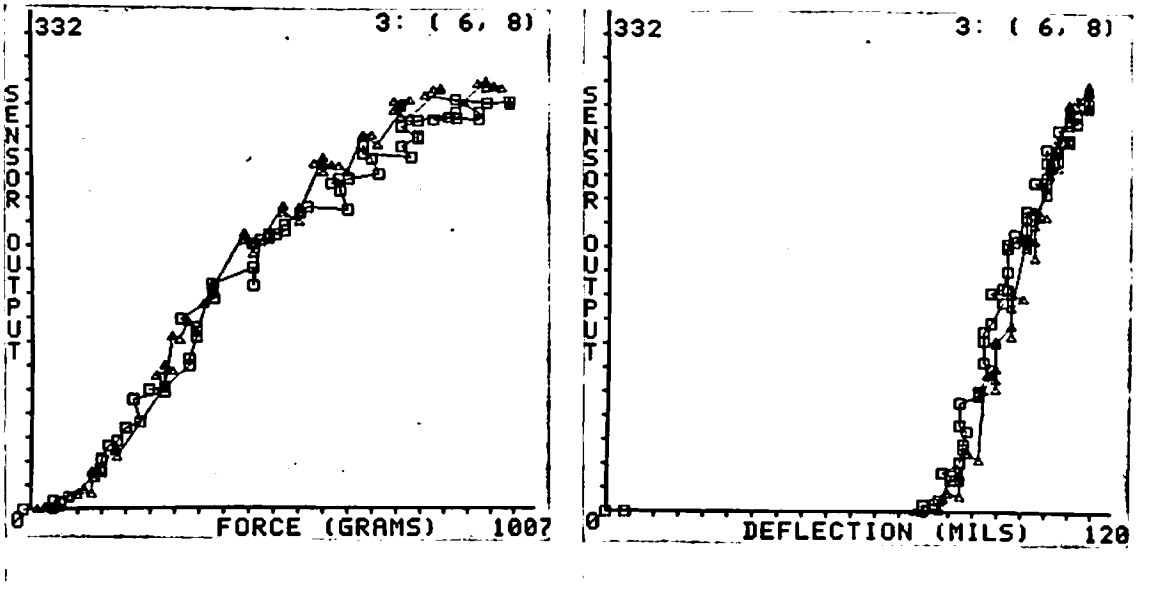
b.


325
2: (10, 2)
SENSOR OUTPUT
0
FORCE (GRAMS)
1086

a.


325
2: (10, 2)
SENSOR OUTPUT
0
DEFLECTION (MILS)
54

b.



a. b.

Figure 3.6

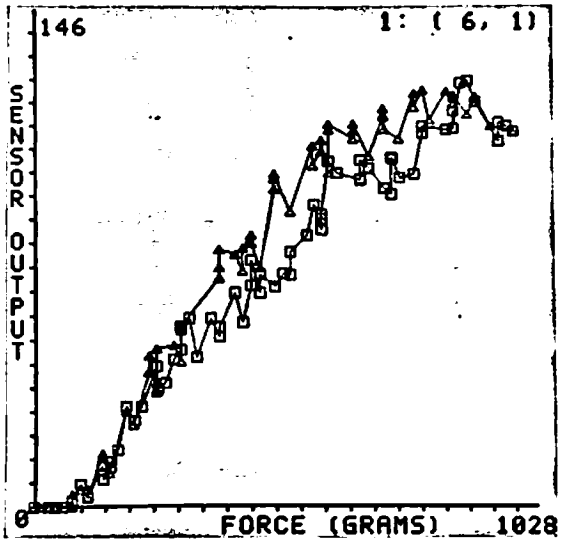
Graphs of the output of three forcels collected by using a probe with the same aspect ratio as a forcel, i.e. 1x2, and twice the area. 3.6a: The applied force measured in grams is displayed along the abscissa while the output of the forcel is measured along the ordinate. 3.6b: This contains the data from the same forcels; however, the abscissas now measure the deflection of the sensor surfaces rather than the applied force. All graphs contain the raw data, see the text.

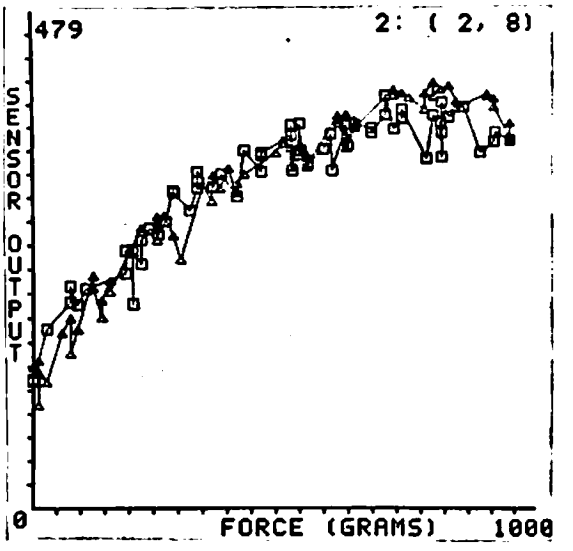
maximum force was attained, the process was reversed and the probe withdrawn in a similar manner. At each force level, the MINC system recorded the output of the transducer being probed, the applied force as determined from the fingertip strain gauges, and the depth on the indentation into the pad. The indentation depth was determined by the vertical movement of the probe from the initial point of contact with the sensor pad. The initial contact point was found by the RS1 for each of the transducers. This was done by lowering the probe in small steps from a position known to be above, and not in contact with, the pad until a preset force threshold was reached. When the threshold was reached, the vertical position was recorded. This position was taken as the height of the surface of the pad; recording began at this point.

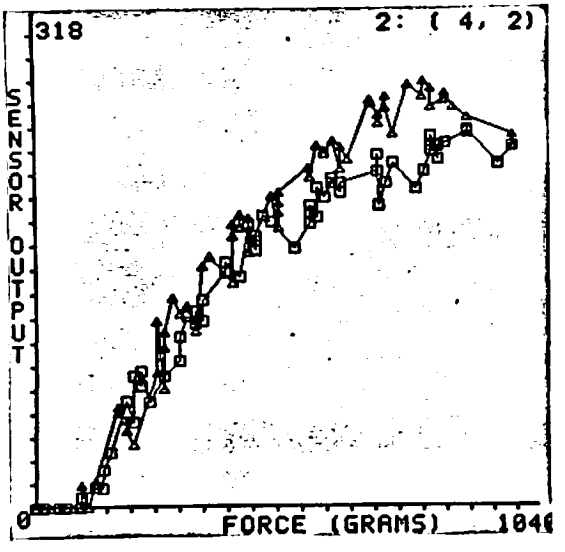
The force threshold used for detecting contact with the surface was inadvertently chosen to be within the range of the random fluctuations of the A/D converter. As mentioned above, these fluctuations were later determined to be as large as 50 grams. This had the affect of causing the system to begin recording before the probe actually contacted the surface of the pad. The recording of the force versus the sensor output was unaffected by this except, of course, for the noise superimposed on the data by the A/D system. Figure 3.7 shows some typical plots of the data obtained in this manner. As described above, the trace is composed of a set of measurements, each indicated by either a triangle or a square. The trace of squares represents the readings as the probe was lowered into the pad while the triangles indicate points obtained as the probe was lifted.

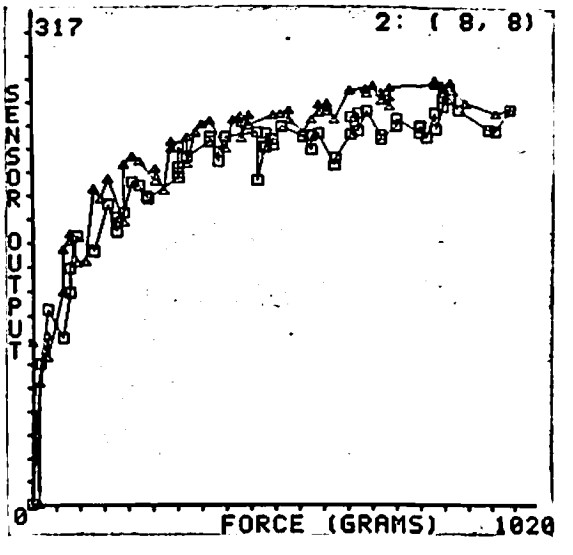
The height versus sensor output, however, was affected since the arbitrarily chosen surface location was used as the relative position of the sensor surface. Figure 3.8a shows the raw data for the trace shown in Figure 3.7 viewed as height along the abscissa. Note that the first reading was taken some distance above the point where the data set appears. Figure 3.8b contains the same data except that the zero deflection point on the abscissa of the graph has been translated to the beginning of the cluster of data points.

The multiple experimental paradigms coupled with the large number of forcels in the three sensors prohibits explicit presentation of all of the response curves. Instead, the behavior of the various forcels is presented qualitatively. The response behavior of the forcels falls into three general categories: non-responsive, confounded, useful monotonic. This last class can be divided further to differentiate between forcels depending upon the quality of the response. Figure 3.9 contains the output graphs from forcels in each of the three classes. The graphs contained in 3.9a shows two good forcels as indicated by a monotonic response. The graphs in 3.9b are from the second class and represent poor responses. The non-responsive class contains all forcels whose output either does not vary at all or whose dynamic range is extremely small. Figure 3.9c contains the output of two such forcels. Notice that there is extreme variation in the output and that the range along the ordinate axis is quite small, 2 bits in one case.


146
1: (6, 1)
SENSOR OUTPUT
0
FORCE (GRAMS)
1028


479
2: (2, 8)
SENSOR OUTPUT
0
FORCE (GRAMS)
1000


318
2: (4, 2)
SENSOR OUTPUT
0
FORCE (GRAMS)


317
2: (8, 8)
SENSOR OUTPUT
0
FORCE (GRAMS)
1020

a.

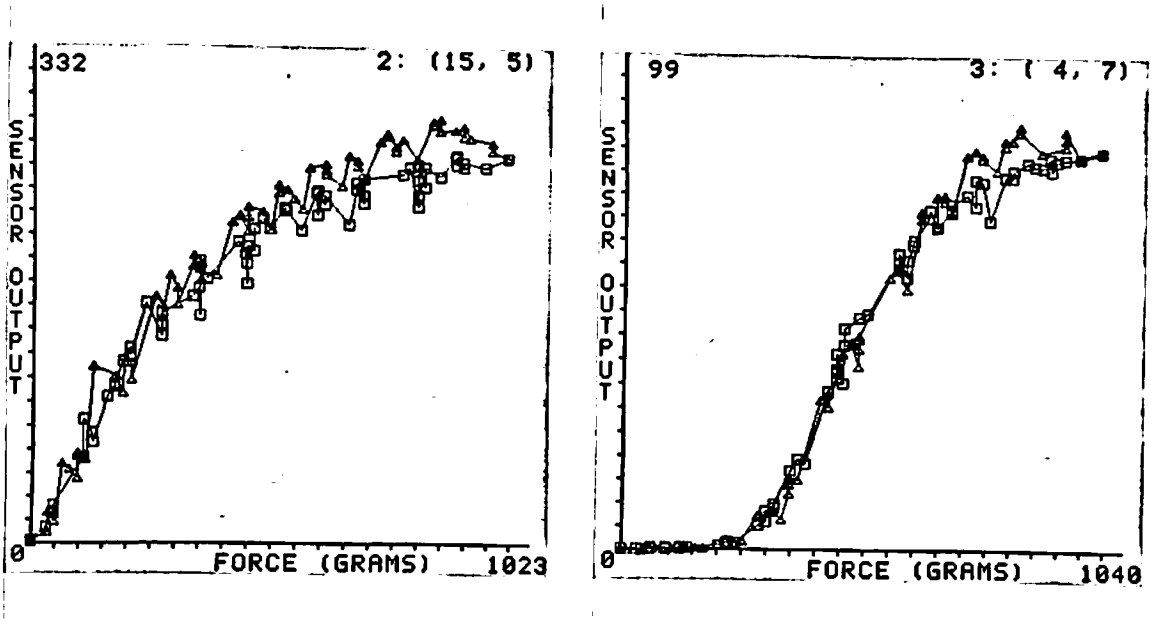
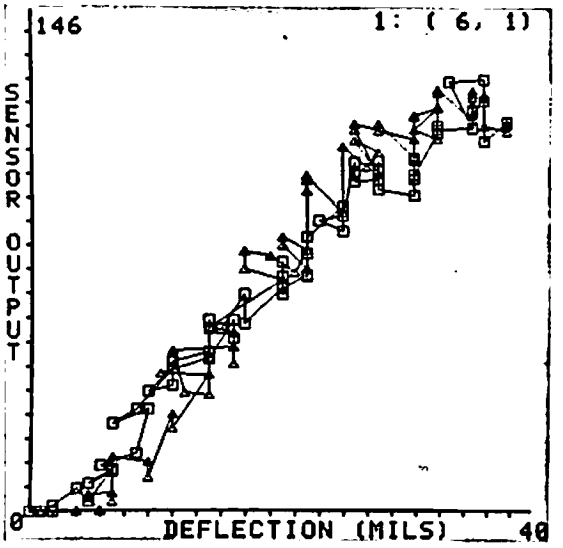


Figure 3.7

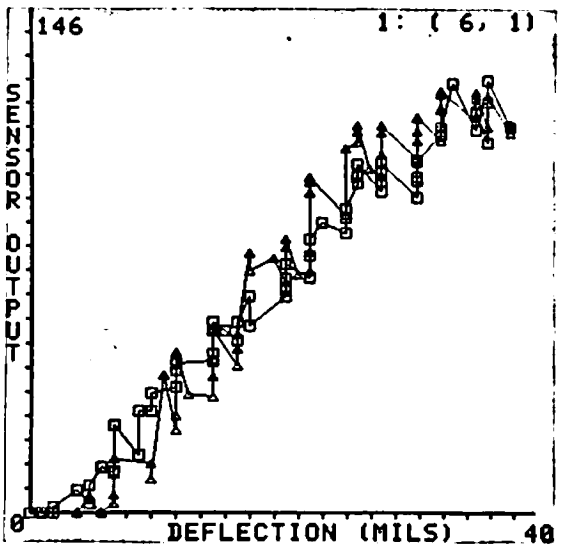
Graphs of Force versus Sensor Output for various forcels in the set of sensors. As for Figure 3.6, the probe used had the same aspect ratio as a forcel, i.e. 1x2, and twice the area. The applied force measured in grams is displayed along the abscissa while the output of the forcel is measured along the ordinate.

Figure 3.8

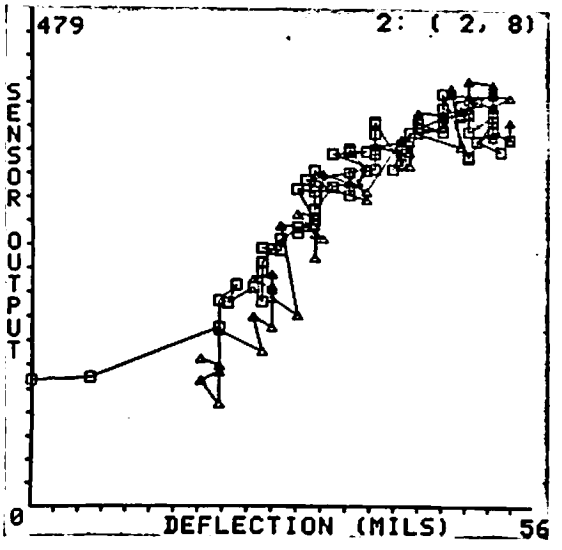
These graphs contain the data from the same forcels as used for Figure 3.7. The graphs plot the deflection of the surface of the sensor against the sensor output. 3.8a: Raw height versus output data. The clustering is due to a problem in determining the exact point at which the probe contacted the sensor, see text. 3.8b: These graphs contain the same data except that the origin of each graph has been translated to the beginning of the cluster.


146
1: (6, 1)
SENSOR OUTPUT
0
DEFLECTION (MILS)
40

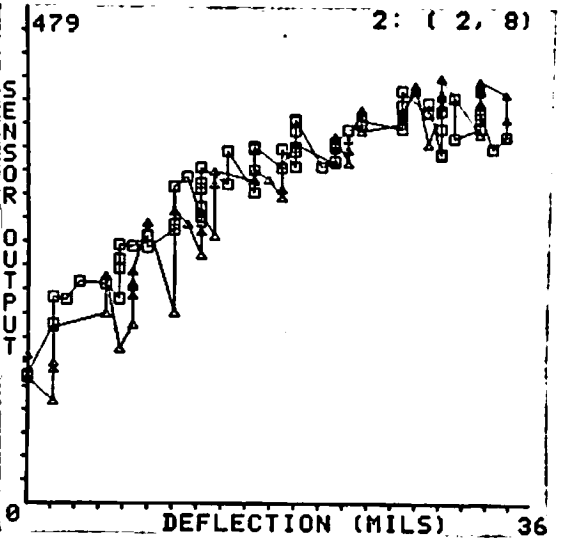
a.


146
1: (6, 1)
SENSOR OUTPUT
0
DEFLECTION (MILS)
40

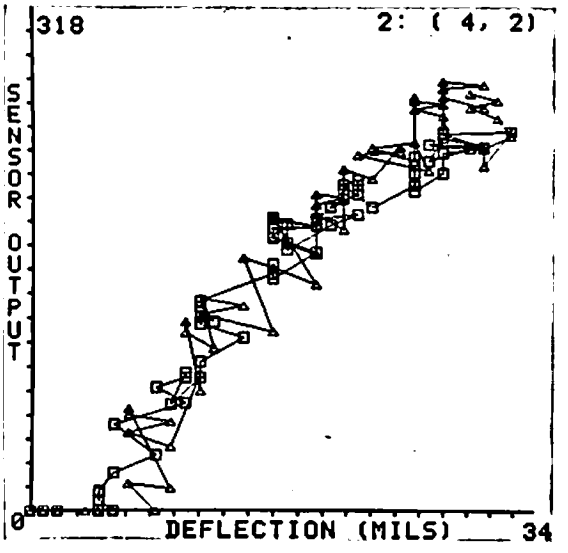
b.


479
2: (2, 8)
SENSOR OUTPUT
0
DEFLECTION (MILS)
56

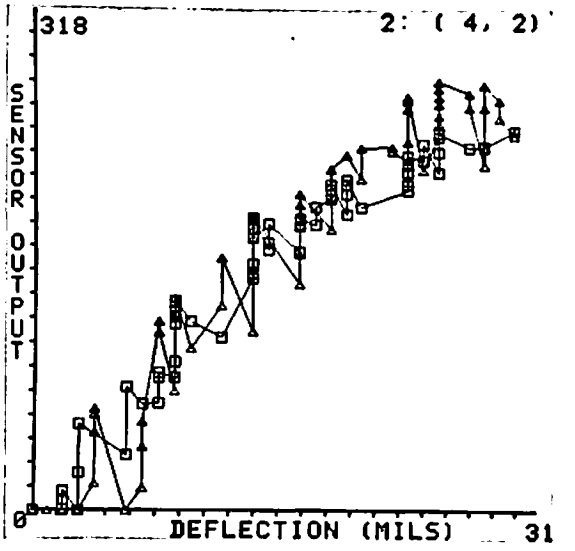
a.


479
2: (2, 8)
SENSOR OUTPUT
0
DEFLECTION (MILS)
36

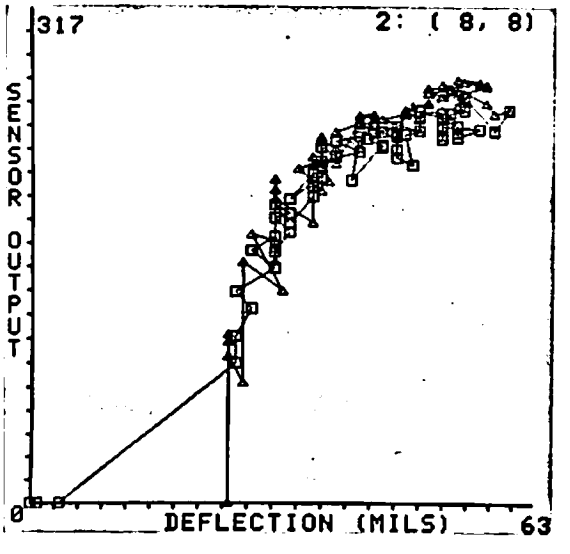
b.


318
2: (4, 2)
SENSOR OUTPUT
0
DEFLECTION (MILS)
34

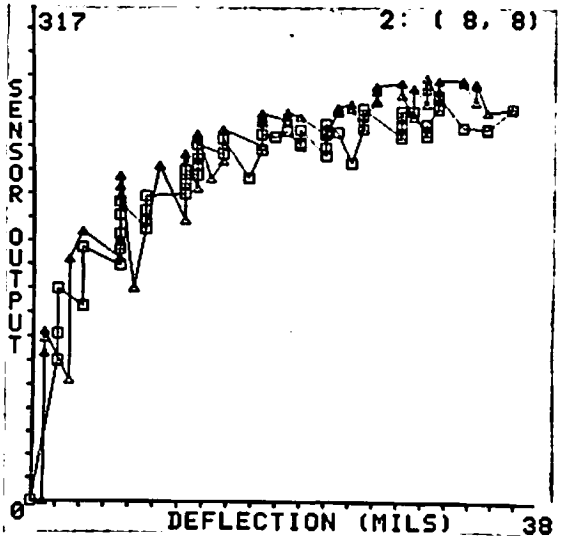
a.


318
2: (4, 2)
SENSOR OUTPUT
0
DEFLECTION (MILS)
31

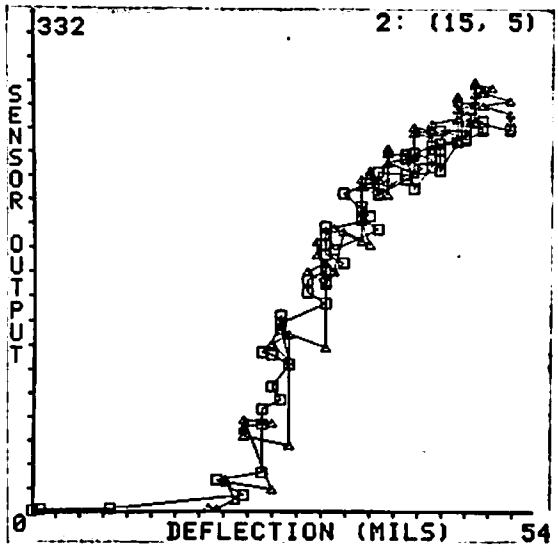
b.


317
2: (8, 8)
SENSOR OUTPUT
0
DEFLECTION (MILS)
63

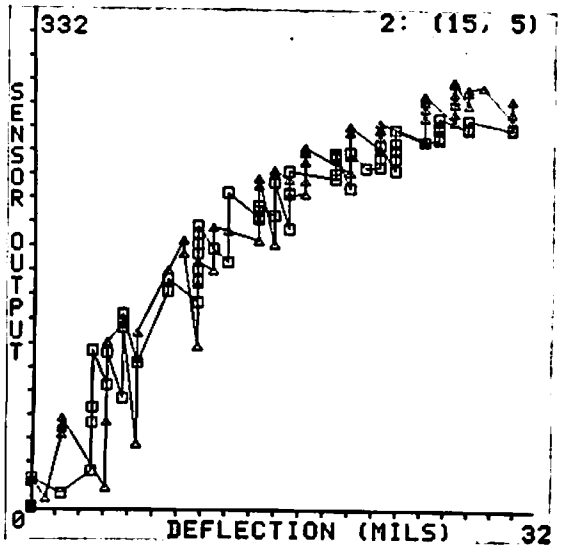
a.


317
2: (8, 8)
SENSOR OUTPUT
0
DEFLECTION (MILS)
38

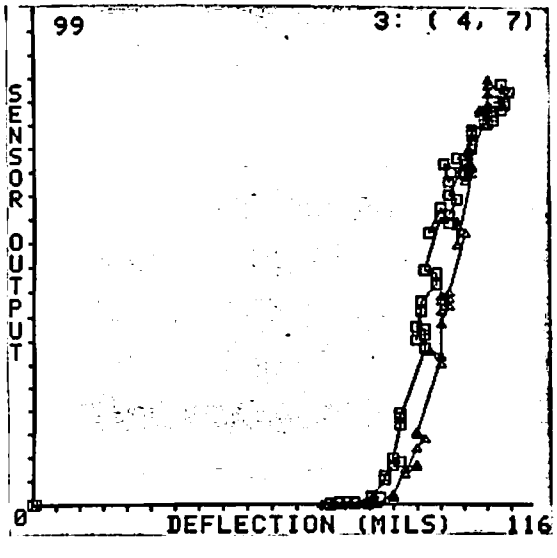
b.


332
2: (15, 5)
SENSOR OUTPUT
0
DEFLECTION (MILS)
54

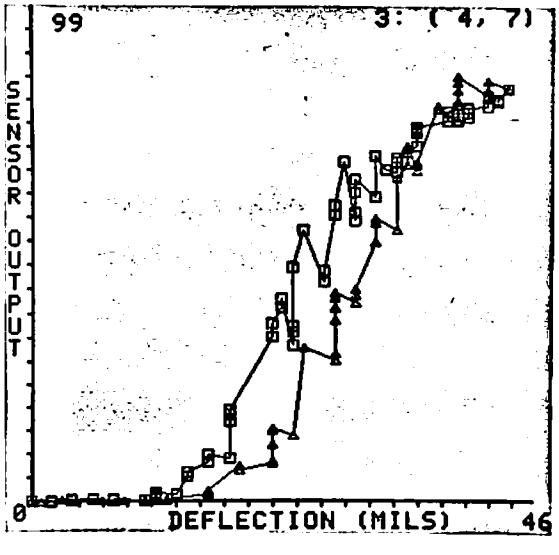
a.


332
2: (15, 5)
SENSOR OUTPUT
0
DEFLECTION (MILS)
32

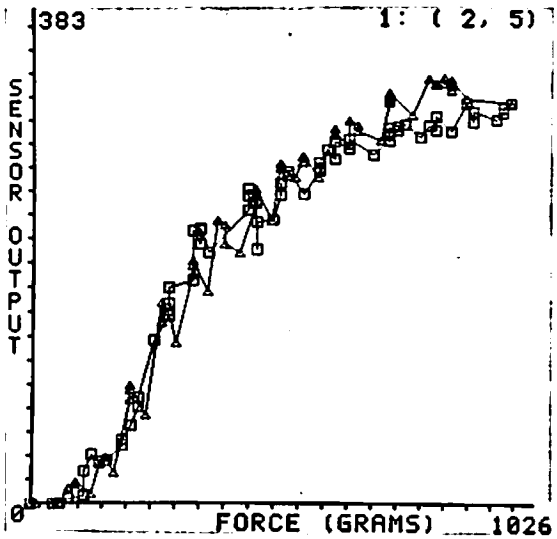
b.


99
3: (4, 7)
SENSOR OUTPUT
0
DEFLECTION (MILS)
116

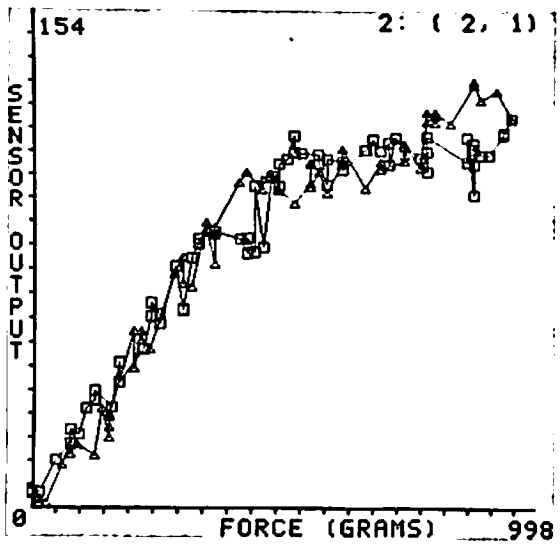
a.


99
3: (4, 7)
SENSOR OUTPUT
0
DEFLECTION (MILS)
46

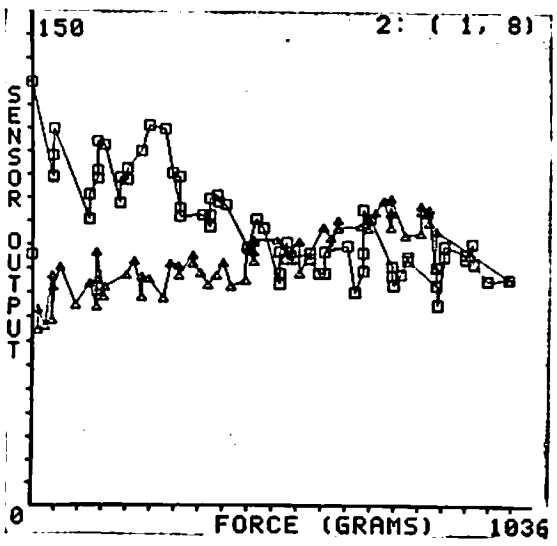
b.


383
1: (2, 5)
SENSOR OUTPUT
0
FORCE (GRAMS)
1026

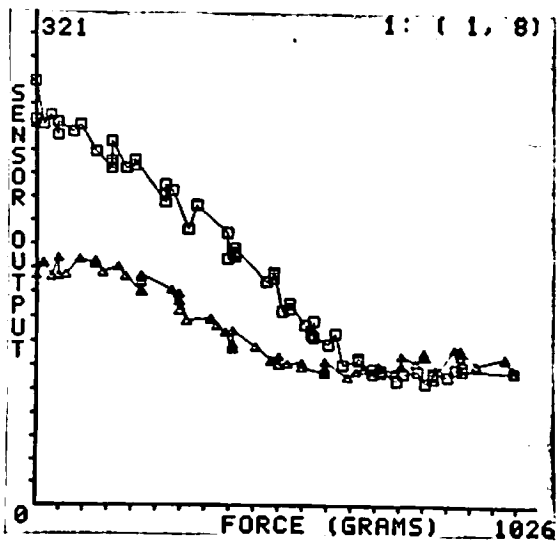
a.


154
2: (2, 1)
SENSOR OUTPUT
0
FORCE (GRAMS)
998

a.


150
2: (1, 8)
SENSOR OUTPUT
0
FORCE (GRAMS)
1036

b.


321
1: (1, 8)
SENSOR OUTPUT
0
FORCE (GRAMS)
1026

b.

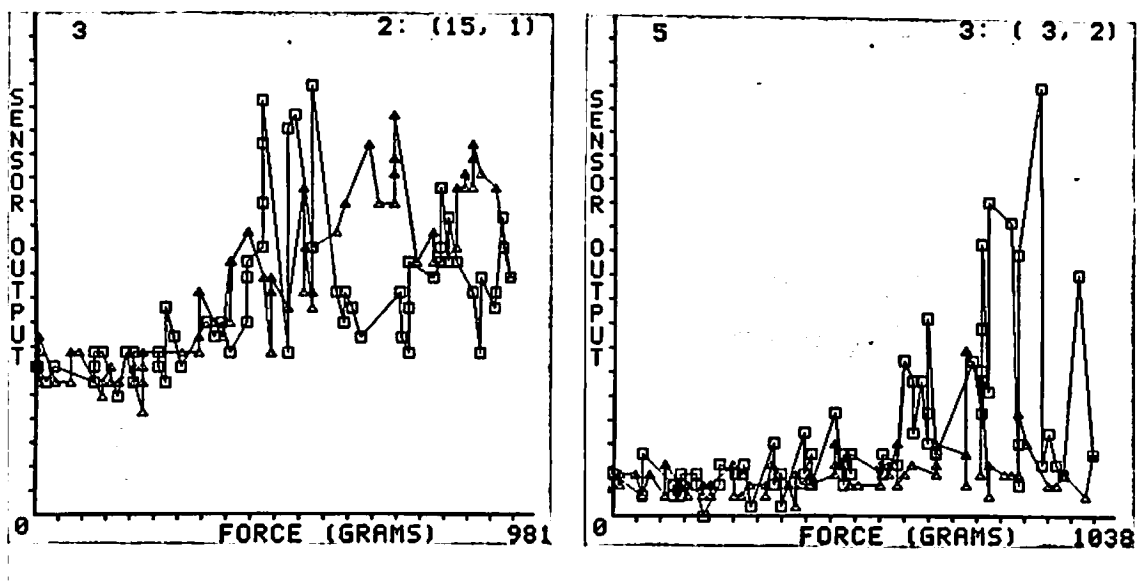


Figure 3.9

This contains the output graphs from forcels in each of the three response classes. 3.9a Contains the graphs from two good forcels as indicated by a monotonic response. The graphs in 3.9b are from forcels with poor or confounded responses. 3.9c Contains the output of two non-responsive forcels.

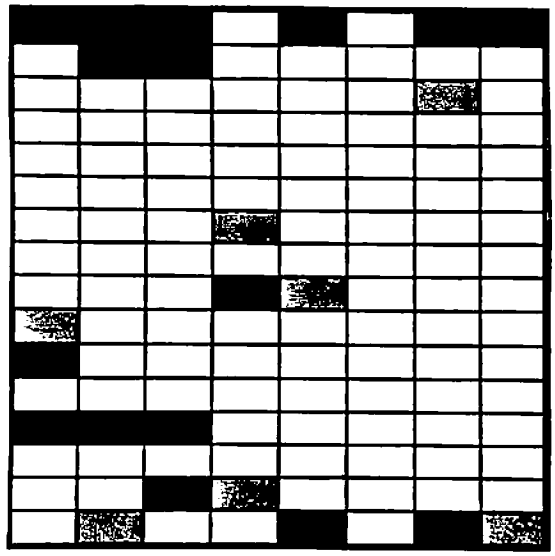
The useful monotonic class actually contains two classes of elements. The first class is composed of those elements which respond monotonically with either a linear, logarithmic, or "s-shaped" form. The second class is composed of those exhibiting a "humped" appearance and are thus not actually monotonic. This response is due, at least in part, to misalignment of the probe with the forcel. This caused strange interactions between the pad and the contact array. These units are included as a subclass of the monotonic useful elements since the information obtained from them is usable over most of the response range.

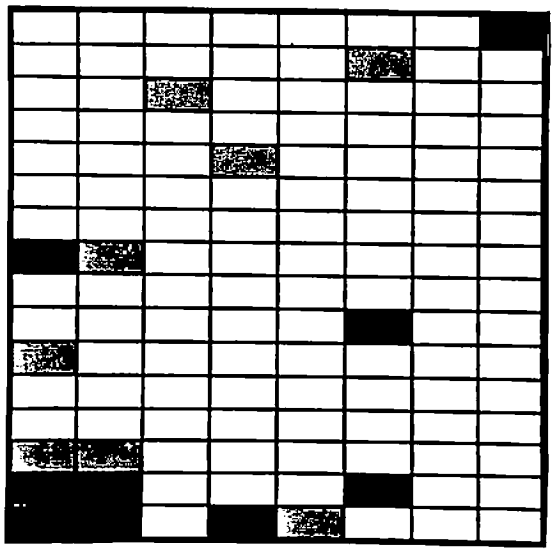
Figure 3.10 contains the response form maps for the three instances of the latest version of the sensor.The response form for a given forcel is indicated by the shade of the corresponding box: white represents monotonic with a useful output range, light grey depicts elements of the second subclass on the monotonic elements (i.e. those actually responding in a "hump-shaped" manner), dark grey indicates those with confounded response forms, and black indicates non-responsive units. It is interesting to note that the highest density of good responders is seen in the centers of the pads. Recall that the sensor pads were aligned with the contact arrays by hand. Rotational misalignment results in the largest errors being in the corners of the sensor: this is seen in the first two sensors, cf. Figures 3.10a and 3.10b. Compression misalignment will appear as non-responsive forcels along one side of the sensor as shown in Figure 3.10c. Pure translational misalignment will appear as uniformly distributed non-re- sponsive units. The total number of forcels in the responding, confounded, and non-responsive classes for the three sensors is: sensor 1 - 112, 4, 12, sensor 2 - 119, 3, 6, and sensor 3 - 104, 1, 23, respectively.

3.2.7 Dynamic Range and Sensitivity

The dynamic range is the range of the input over which the sensor output is useful. "Useful" here is not equivalent to monotonic since the response of a good forcel tends toward some maximum value asymptotically; rather, "useful" is intended to mean that two input stimuli can be differentiated based upon the sensor output. Sensitivity is the measure of how small an input stimulus can be detected from the sensor output. Of course, the noise present in the output confounds the problem as does the hysteresis. In addition, the dynamic range and sensitivity vary for each forcel in the array and no single value is correct for even a reasonable fraction of the forcels.

The response form experiment provides the most appropriate paradigm for investigating these levels. Certain forcels respond to extremely light forces and deflections, i.e. on the order of less than 10 grams or less than a few tenths of millimeters. Other forcels, however, do not show a response until several tens of grams are applied or deflections near one millimeter are reached. These inconsistencies are due to fabrication variances over the area of the sensor pad and would be reduced if an automated manufacturing process were to be employed.





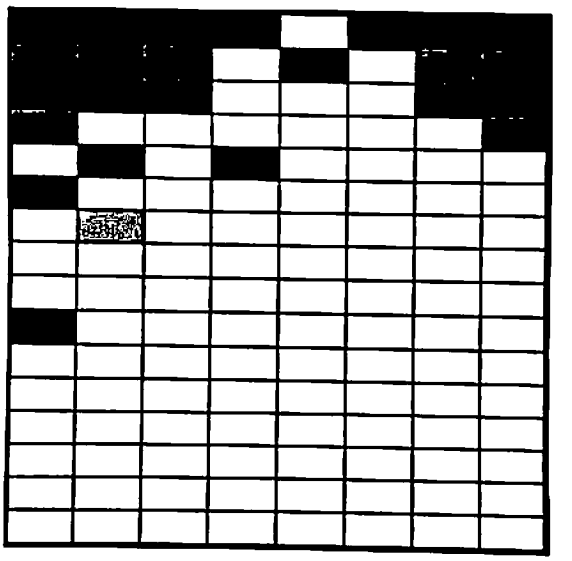


Figure 3.10

These figures contain maps of the response types of the forcels in three sensors used during the response testing. Each rectangle represents a forcel in the sensor: the shade of the rectangle indicates to which response class it belongs. The white elements fall into the class of good responders, i.e. linear, logarithmic, or s-shaped responses. The light grey represents the hump shaped responders which are also considered as good responders in Figure 3.9. The dark grey elements exhibit a confounded response. That is, they do respond, yet the response cannot be characterized as those in the previous shades can. The black elements are non-responders.

At the other end of the spectrum, the testing was ended when a force of 1500 grams was reached in the case of the probe with twice the area of a forcel. This corresponds to an equivalent total forcel of 96 Kg per square inch, or roughly 211 lbs per square inch. Good forcels showed response curves at this level which had flattened out, but not to the point of being useless. However most of the responding forcels reached a plateau level near the 1500 gram mark. As with the sensitivity threshold, the upper discernible limits varied from one unit to the next. During early testing on the Instron materials testing machine, forces up to 400 lbs were applied uniformly over the surface of the sensor. At this level the forcels had essentially all reached output levels where slight variations in the stimulus were non-discernible from the output. The transducers returned to their normal response characteristics once the stimulus forces were reduced to reasonable levels. This showed that the device could withstand an overload without permanent damage.

3.2.8 Noise

Noise can be defined as the uncontrolled variation in a signal due to unmodeled characteristics of the device or affects of phenomena of the environment not related to the characteristics being measured. In the case of the tactile sensor, noise appears as the variation in the output of the transducers given a constant stimulus. In the simplest case, the output of the sensor can be monitored with no force applied to the device. The individual transducer outputs typically vary by one or two bits when the full range of the sensor is represented by 10 bits. Unfortunately, this may not be a representative case for the sensor since, as seen from the discussion of the response forms, there is usually a dead force region where there is no response to changing force.

A more useful case occurs when a constant, non-zero stimulus is applied to the sensor. Fluctuations in the output under this paradigm are more representative of the noise. Theoretically, this is exactly the experimental paradigm used during the response form recordings. Observing Figures 3.6, 3.7 and 3.8, one sees a number of vertical clusters in the recordings. Each cluster corresponds to a set of readings taken at the same force/deflection. The variation in the output for a given force is approximately 10% of the output range. The fluctuations in the A/D conversion hardware on the RS1 imply that the locations of the data points along the abscissa are not accurately known. In fact, the variation of +-50 grams is roughly 10% of the range in the case of the 2X probe. Therefore, the variations seen in these graphs are due in part to noise in the sensor and in part to the problem in the A/D hardware.

Another paradigm involved the application of a a given arbitrary stimulus to the sensor and recording the images over time. The sensor pad relaxes slightly over time as it moves around on the contact matrix; thus, the constant application of a stimulus implies actively controlling the system based upon the total applied force. Rigidly holding a stimulus relative to the pad proved insufficient for obtaining non-confounded results. Fluctuations in the output appeared to be about twice the magnitude of the fluctuations present without

any stimulation.

A contributing factor to the actual noise in the sensor is the length of the wires which run from the sensor to the scanning and conversion electronics. In the present design, these leads are neither shielded nor in the form of twisted pairs. Future designs, which employ a combination of shorter leads and shielding or twisting of the leads, will exhibit somewhat less noise.

In summary, it can only be stated that the spontaneous variation in the sensor output is equal to no more than 10% of the range of the device. Further, the variation in the output when no load is applied is on the order of 1-2% of full scale. However, the experimental setup and the difficulties involved in controlling the application of a stimulus to the sensor precludes finer evaluation.

3.2.9 Hysteresis

Hysteresis can be thought of as the delay in the change of the output of a system given a change to the input. In the case of the tactile sensor this can be thought of as having two cases. First, how long does it take for the sensor to respond when an instantaneous change stimulus is applied? Similarly, how long does the output remain at a given level when the stimulus is removed or changed?

Some insight into this characteristic of the sensor can be gained from again considering Figure 3.7. A median filter was used to remove the severe oscillations in the two traces; results of this filtering with windows of different widths are shown in Figure 3.11.

Figure 3.11d contains the results after using the widest window and shows the hysteresis in the sensor. One observes a monotonic increase in from the point of initial response to the point just below the maximum. (The dip at the very end of the graph is due to the method by which the filter window was applied.) However, as the force is removed, the output remains high for some period of time. The units in the figure would indicate that this time is on the order of seconds; however, in practice this interval was never observed to be that great. The difference between the curves represents the hysteresis.

Testing of samples of the RTV11 silicone rubber used in the pad on the Instron machine revealed that the material itself exhibits no hysteresis within the limits of the testing apparatus. This implies that the observed hysteresis is due to two things. First, the electronic scanning and conversion hardware must contribute, a fact which is believable given the discussion on the settling time of the hardware. Secondly, the electro-mechanical characteristics of the contacts between the rubber pad and the contact array must provide a contribution. The latter phenomenon is not understood well due to the complex nature of the interaction.

A question more important than the magnitude of the hysteresis concerns the difficulties it poses in the processing and interpretation of the data. The amount of hysteresis seen is small relative to that observed in certain nerve responses in the human tactile system. Furthermore, the mechanical response of the rubber pad is faster than that of human skin. Philosophically one could

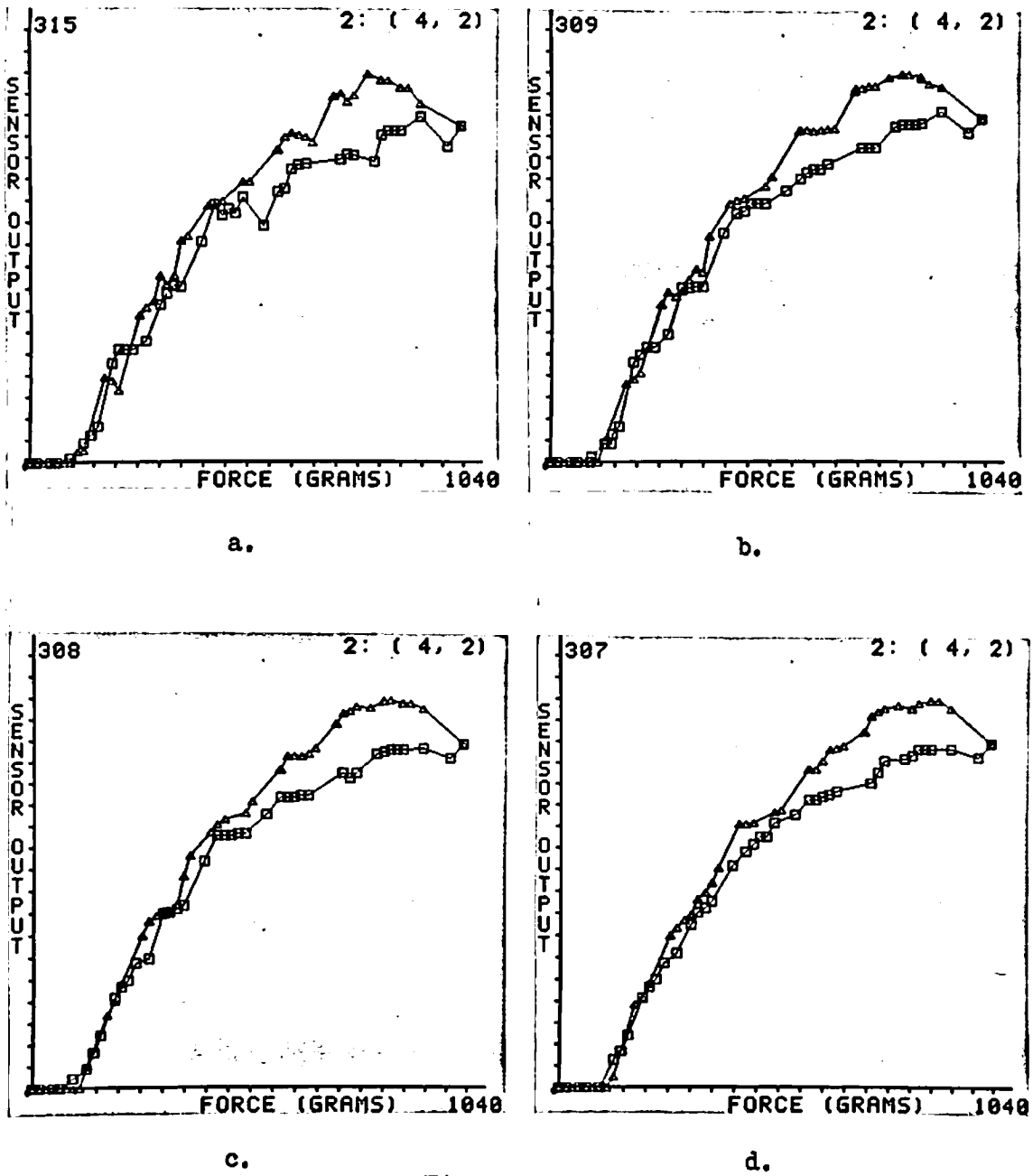


Figure 3.11

This contains graphs of the data from the forcel shown in Figure 3.7a after filtering with a median filter of different widths. 3.11a: the filter window width used was 25 grams. 3.11b: 50 grams. 3.11c: 75 grams. 3.11d: 100 grams.

argue, then, that proper processing techniques can utilize data from a hysteretic device. In fact, the processing discussed in the succeeding chapters does not attempt to model the hysteresis of the sensor but rather views it implicitly as part of the images with quite reasonable results.

3.2.10 Point Spread Experiments

The previous discussions have shown the response forms seen in the forcels in the sensor pad. The intent of this section is to model the sensor as a physical device and relate the results of this modeling to the observed behavior of the sensor. Recall that two basic parts comprise the sensor itself: the rubber pad and the mating PC board. The rubber pad consists of a matrix of loops made of a carbon-filled silicone rubber embedded in an insulating supporting body of silicone rubber. The PC board contains the matrix of electrodes which contact the conducting loops.

As pressure is applied to the pad, three resistances change. These include the internal resistance of the carbon-filled column and the contact resistances at the interfaces between the loop and the PC board. The contact resistances drop as the pressure increases. In the mathematical modeling of the device these resistances are modeled as one lumped resistance. One purpose of this section is to investigate the decoupling of these terms.

The object of this section is the rubber pad and its interaction with the underlying surface. The pad can be considered as a block of homogeneous material of known dimensions. The differences between the compositions of the columns and the supporting rubber are so slight that they can be ignored here. The ANSYS engineering analysis system was used to produce a finite element model of the sensor pad. The model was used to simulate the exercising of the pad under circumstances analogous to those under which data images were collected from the sensor.

The characteristic of interest is the degree to which a sharp edge stimulus is blurred in the resulting force image. In a more general sense, given a point stimulus, what is the spread function of the sensor? Such information is useful in bounding the class of edge and point stimuli which are discernible by the sensor. The approach taken began by producing an ANSYS finite element model of the sensor pad, simulating the stressing of the pad with an indenter, and observing the deflection and pressure distribution in the pad. These data were then compared with the experimental results obtained from a real sensor poked with a real indenter.

The ANSYS software package is a system designed to allow the user to model mechanical structures in order to analyze various properties. The system can analyze static and dynamic characteristics, plastic and elastic creep and swelling, and deflections and force loading as well as heat transfer properties and current flow. The software contains extensive model creation facilities, simulation modules, and post-simulation processing software for analysis and presentation. Finite element idealization of the mechanical structure of interest is used throughout the modeling process. For further information, the reader is

referred to [52]. This system was used to produce a model of the rubber pad which was then exposed to deflections analogous to those present in the sensor during a particular controlled situation.

Axisymmetric units were used to model the pad. While the actual sensor pad consists of an array of carbon-filled silicone rubber structures embedded in an insulating silicone rubber matrix, these details were not modeled due to the complexity. The pad was considered to be a homogeneous block of material. Figure 3.12 contains a schematic view of these elements in relation to the actual sensor pad. The axisymmetric units are annular in appearance. The use of these units greatly simplifies the generation of the mesh. The simulations employed a cylindrical stimulus applied to the center of the pad, thus the influence is uniform in all direction. Since this is the case, it is sufficient to consider as output the information regarding the elements intersected by the half plane defined by the positive x-axis and the y-axis. Figure 3.13 contains the element structure along this cross section.

Each square in Figure 3.13 represents the cross section of one of the annuluses. The overall width of the cross section of the modeled pad is 0.5 inches with a height of 0.2 inches. Thus the modeled pad is a cylinder 1.0 inches in diameter and 0.2 inches thick. The actual pad used on the sensors is a 1.0 inches square 0.2 inches thick. The pad model consists of 232 annuluses arranged in 8 disks of 33 rings per disk. Since the probe was to be applied at the center of the pad, the rings near to the center are much smaller in width than those in the periphery. The inner 28 columns are composed of annuluses with cross sectional dimensions of 0.01 inches in width and 0.025 inches in height. These rings correspond to the central 0.28 inches of the pad, measured radially. The remainder of the pad model is composed of 5 columns of rings whose cross sections each measure 0.044 inches in width and 0.025 inches in height.

The stimulus used in the simulation is a cylindrical probe with a radius of exactly 0.03 inches. This corresponds to the first three columns of rings in Figure 3.13. These mesh and probe configurations were used to simulate the case where a probe was deflected an absolute amount. The modulus of elasticity of the RTV11 silicone rubber used in the actual sensor pads was not available from the manufacturer and so tests on samples of the material were conducted. The empirically determined modulus of elasticity was 300 psi. Also, a poison's ratio of .485 was used for the material. Figure 3.14 shows four views of the cross section of the pad with the probe deflected 0.01, 0.02, 0.03, and 0.04 inches respectively.

The mesh in the figure is displayed as in Figure 3.13. That is, only the positive x half plane of the cross section is shown. Figure 3.14 illustrates two interesting points concerning the sensor pad. First, the response of the pad at the edge of the deflection body is one of a smooth change from full deflection at the edge of the probe to no deflection some distance away. Secondly, the deflections in the surface die out a short distance from the stimulus. Figure 3.15 contains a graph of the deflections of the top row of nodes in the mesh with the scale expanded to expose the details. There is one curve on the graph for

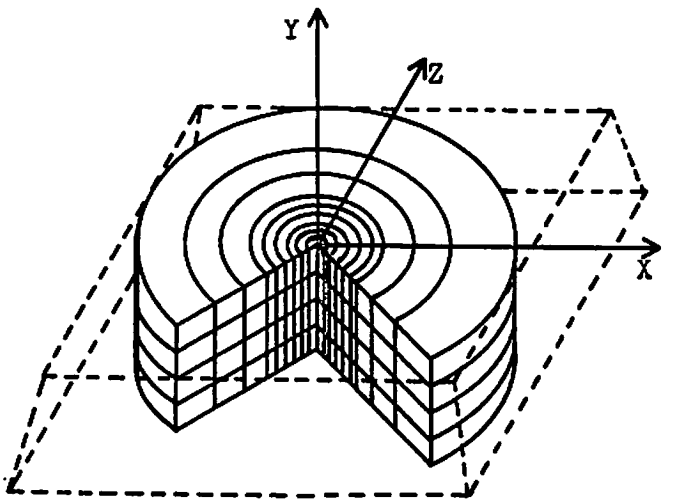


Figure 3.12

This figure contains a schematic of the finite element model used in the ANSYS simulations displayed in relation to the sensor pad used in the version 3 sensor. The model is constructed from annular elements causing the overall structure to appear as a disk. The cutaway shows the detail of the annular segments. The model is shown with the outline of the raised section of the sensor pad, cf. Figure 2.24.

Figure 3.13

This contains a cross sectional view of the annular model elements. The cross section was taken along a radius of the model with the left edge of the figure corresponding to the exact center of the pad. Each box represents the cross section of one of the annuluses. The pad consists of a total of 232 annuluses.

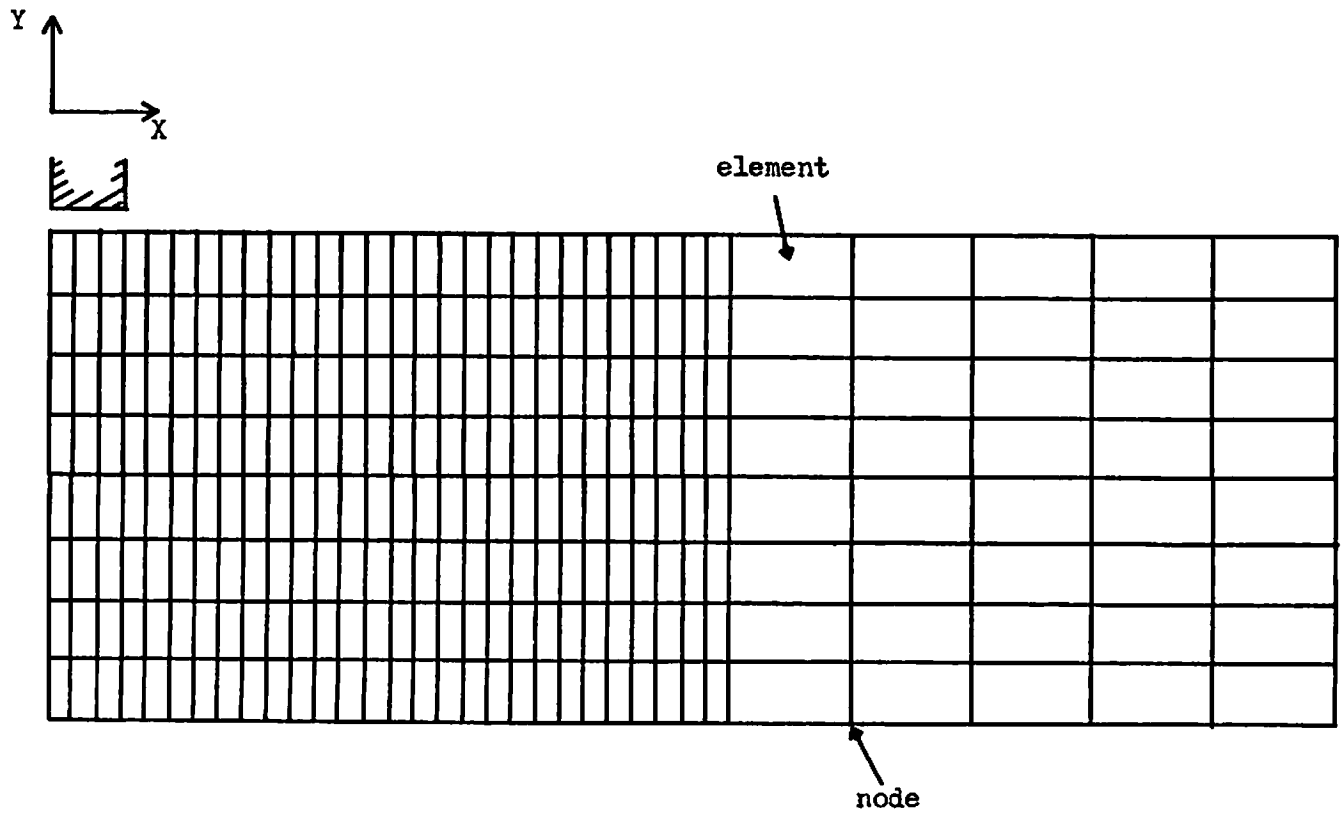
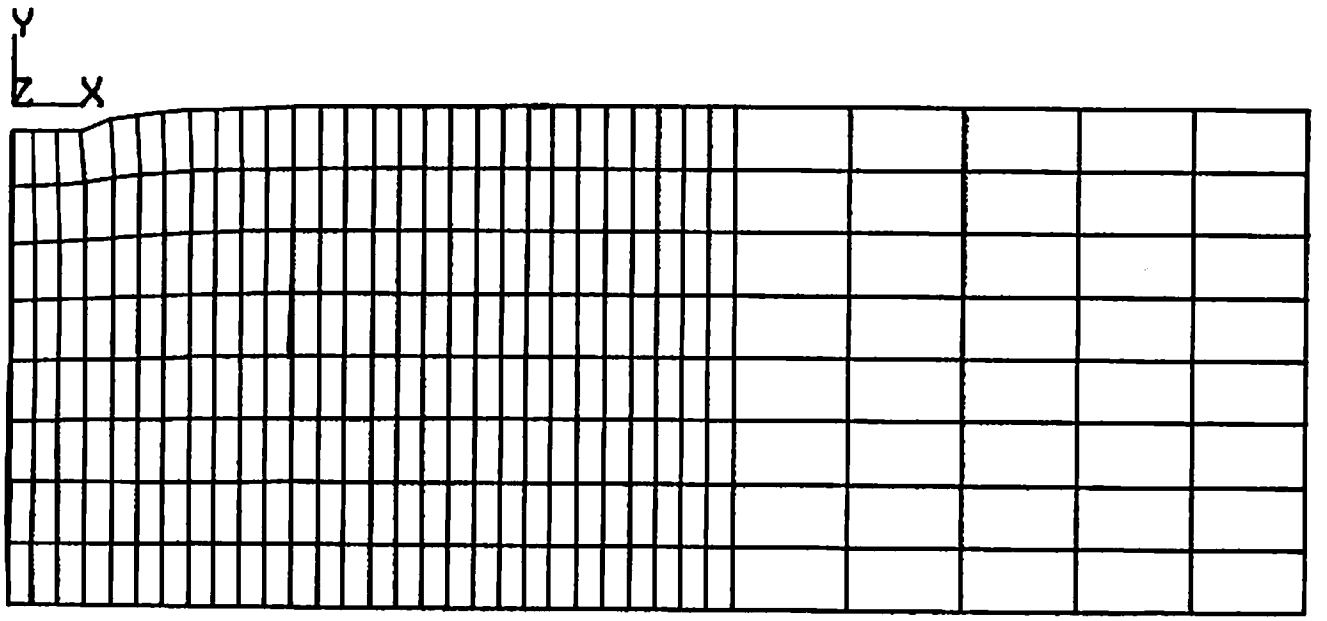
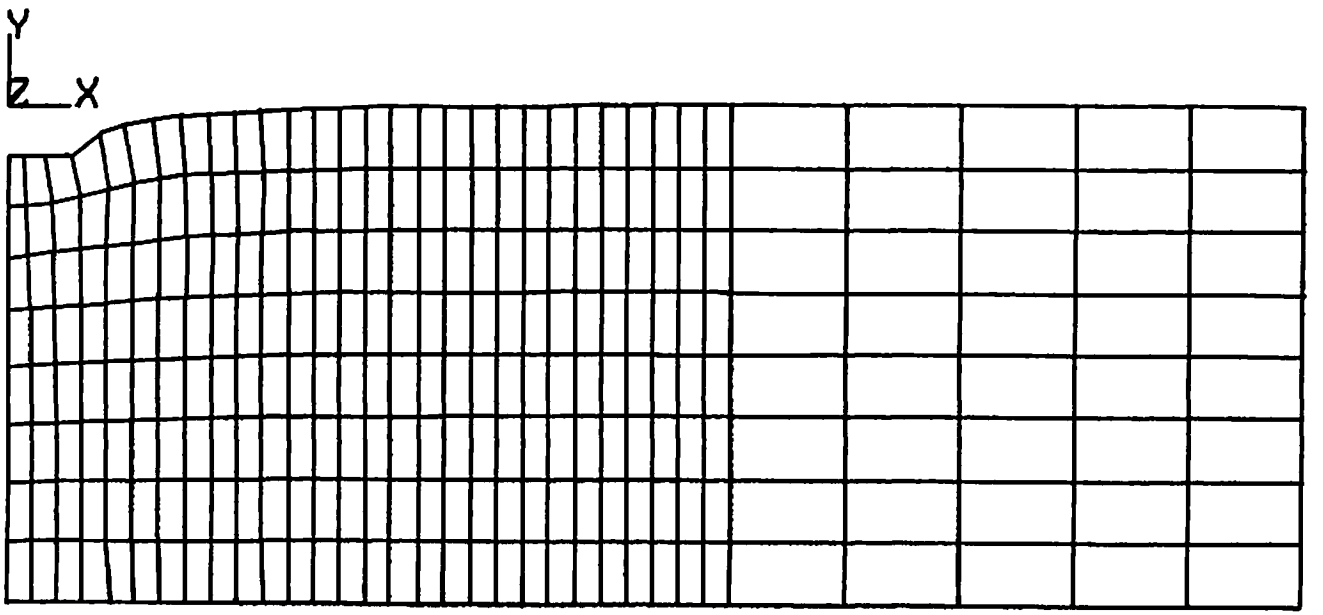

Y
X
element
node

Figure 3.14

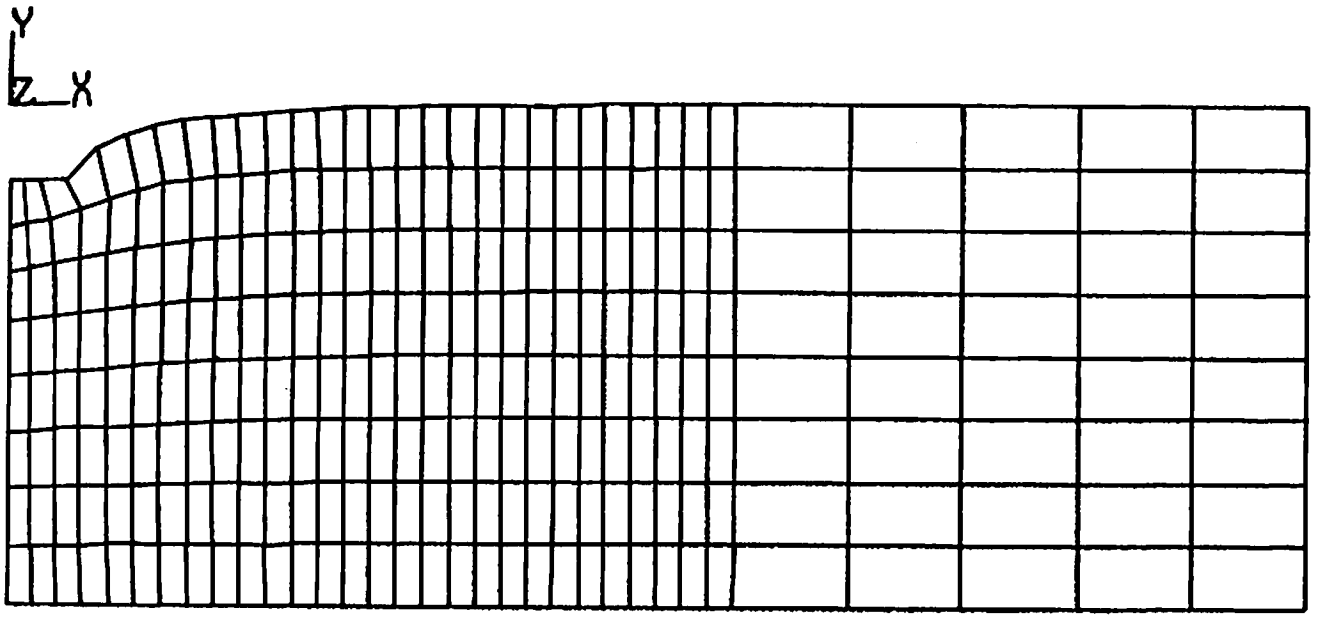
These four graphs show the deflection of the mesh comprising the ANSYS model for three different deflections of the indenter. The indenter was modeled as having a cylindrical shape and a radius equal to the width of the inner three annuluses. 3.14a: indenter movement of 0.0105 inches. 3.14b: 0.0209 inches. 3.14c: 0.0314 inches, 3.14d: 0.0418 inches.



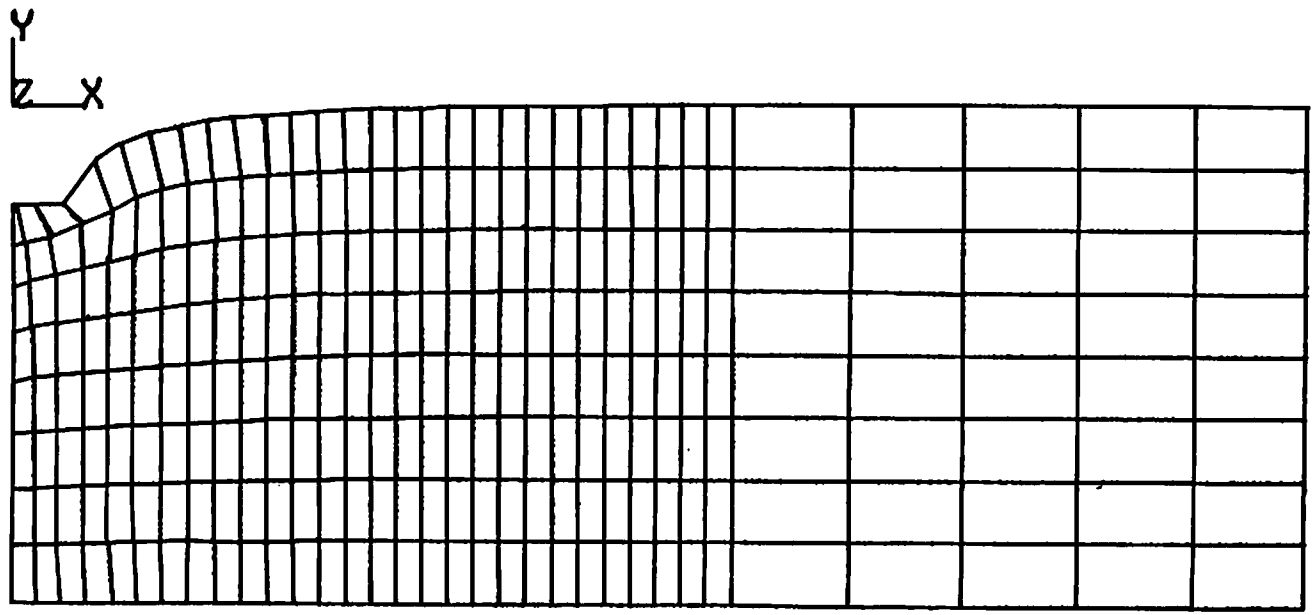
a.



b.



c.



d.

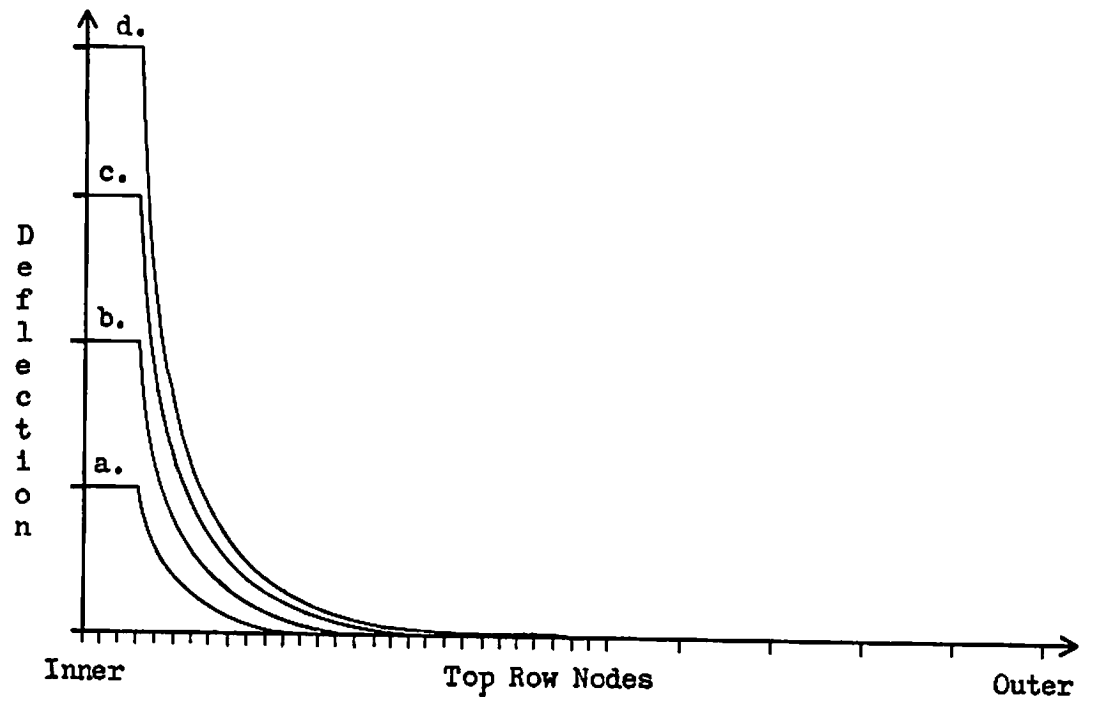


Figure 3.15

This is a graph of the deflections of the mesh nodes along the top row of the model. The deflection is plotted for each of the nodes along the top of the mesh in Figure 3.13. The four traces represent the graphs for the four deflection cases in Figure 3.14: a - 0.0105 inch indenter deflection, b - 0.0209 inches, c - 0.0314 inches, d - 0.0418 inches.

each of the deflection curves in Figure 3.14.

An important feature of these data is the distance on the sensor pad required for the material to recover from the deflection. That is, how does the deflection due to a step stimulus spread over the surface of the pad? This "spread function" is of interest since it represents a blurring of sharp edges in tactile images. The recovery of the material assumes a logarithmic form with recovery to 50% of the initial deflection within 0.02 inches of the probe edge. For all cases, the deflection of the sensor pad is less than 1% of the probe deflection within 0.17 inches of the edge of the probe. It should be noted that while these data agree with the observed behavior of the material in form, the actual material did not recover in as small a spatial extent as the simulation predicted. This was due to the inaccuracies in the model and inconsistencies in the actual pads. The size of the forcels is compared with this spread function below.

Unfortunately, the resistances being measured by the sensor are not

indicative of the state of the top of the pad. Recall that the quantity being measured is a resistance which is actually the sum of three resistances: the loop resistance and two contact resistances. The resistance of the loops of carbon-filled rubber is a function of the compression of the material, i.e. a characteristic of the internal sections of the pad. The contact resistances are functions of the forces present at the interface between the bottom surface of the pad and the rigid supporting board. The pressures exerted on the hypothetical surface supporting the model by the sections between the nodes along the lower edge of the mesh can be graphed. Such graphs (not shown here) represent the pressures seen at the interface between the sensor and the underlying PC board for the four indenter cases shown in the previous figure and are indicative of the pressures at the contacts between the loop and the board. The maximum pressure occurs directly under the indenter and the values decrease exponentially with increasing distance from the indenter.

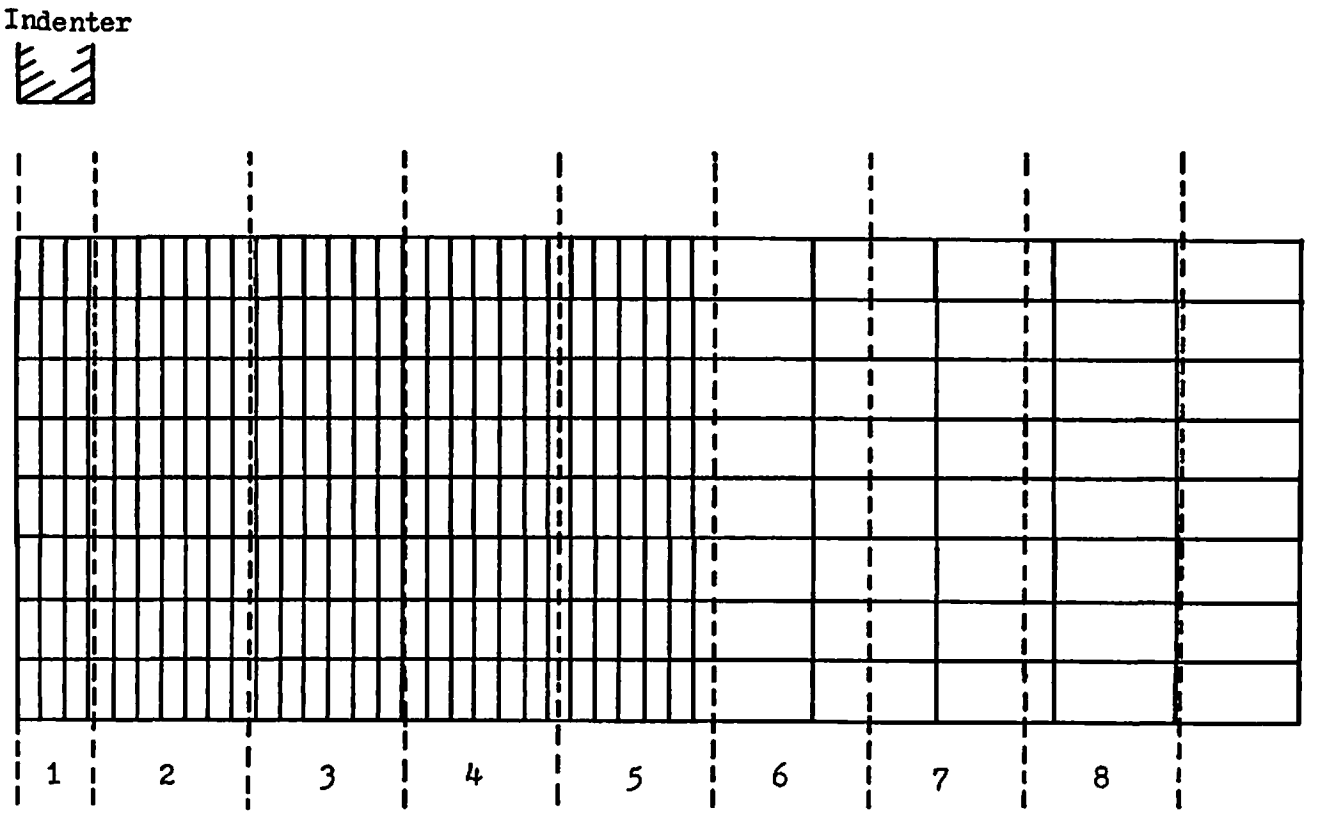
Figure 3.16 shows the relationship of the elements in the ANSYS model to the forcels in the sensor. The innermost forcel is centered under the indenter; thus, in the figure this forcel is only half the width of the other forcels. Also, the section through the sensor was taken along the digital axis. That is, the cross section is along the axis of highest density of transducers. There are, therefore, sixteen forcels in the cross section and since only half of the sensor is shown, only eight elements are shown.

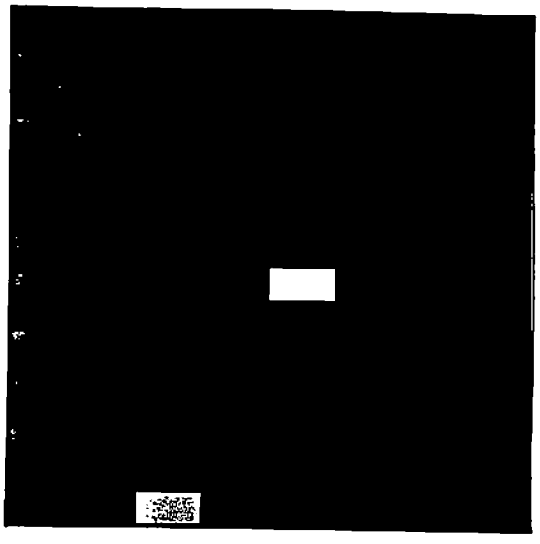
Let us now return to the tactile array sensor data. An experiment was performed with the intention of understanding the spread function of the sensor. It is interesting to compare the experimental results with the simulation output. The data collection technique used was similar to that employed to retrieve the response curves of the individual forcels. An IBM RS1 manipulator was programmed to probe a selected element while a DEC MINC system recorded the tactile images. Three different probes were used, each having a cross sectional area with a 1x2 aspect ratio. These probes, along with the probe holder, are shown in Figure 3.5 along with the proboscis used in the uniform force experiments. The areas of the tip were one half the area of a forcel, the area of a forcel, and twice the area of a forcel resulting in dimensions of 0.0442"x0.0884", 0.0625"x0.125", and 0.0884"x0.1768" respectively. The probes bolt to the holder which, in turn, is held by the fingertips of the RS1 gripper.

Figure 3.17 shows the tactile images resulting from the probing. Figure 3.17a was produced by the smallest probe, i.e. the probe area equaled one half of the area of a forcel, 3.17b by the medium probe, and 3.17c by the largest probe. Each image has been "corrected" using a technique chosen in the image correction section, see below. It is seen that the correction technique is not perfect as evidenced by the forcel at the bottom of the third column. Notice in all cases that the forcel over which the probe was centered produces the greatest response with the magnitude of the activity decreasing radially from the probed forcel. The amount of force applied with the three probes varied with 500, 1000, and 1500 grams being applied by the small, medium, and large probes respectively. The images in the figure have been scaled so that the largest value

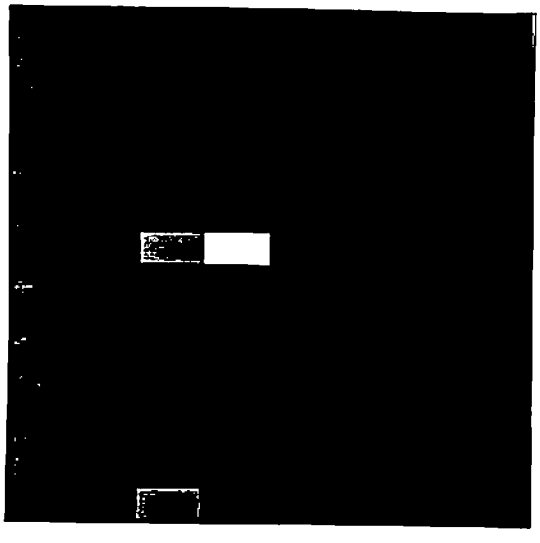
Figure 3.16

This shows the relationship between the annular elements in the ANSYS model to the forcels in the sensor pad. The model has been aligned with the sensor forcels so that the indenter impinges upon only one forcel. Thus the forcel corresponding to the left edge of the pad is shown as half the width of the other forcels.

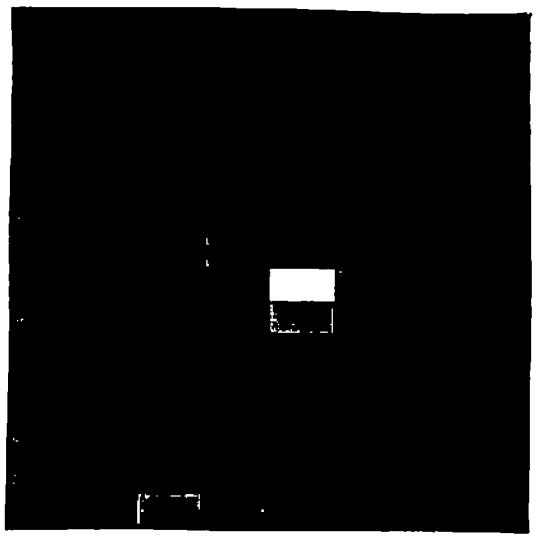

Indenter
1
2
3
4
5
6
7
8



a.



b.



c.

Figure 3.17

This figure shows the tactile images resulting from probing with the single forcel probes described in the text. 3.17a was produced by the smallest probe, i.e. the probe area equaled one half of the area of a forcel, 3.17b by the medium probe, and 3.17c by the largest probe. Each image has been "corrected" using a technique chosen in the image correction section of this chapter. Notice in all cases that the forcel over which the probe was centered produces the greatest response with the magnitude of the activity decreasing radially from the probed forcel. The amount of force applied with the three probes varied with 500, 1000, and 1500 grams being applied by the small, medium, and large probes respectively. These images have been scaled so that the largest value in the image is displayed as full white.

in the image is displayed as full white.

Figure 3.18 contains a slightly different view of the data in Figure 3.17. These displays are graphs of the output of the sensor taken on a slice through the tactile images. The column containing the probed forcel was selected to provide the data for each graph. In order to represent the data in a manner similar to Figure 3.15, the output values at a given distance on either side of the probed forcel were averaged and the trace on only one side of the probed forcel displayed. The left most forcel in the graph represents the probed forcel in the tactile image. Each value to the right of the center forcel represents the average of two forcels, one above and one below, that number of forcels from the central one. In each case, the response declines exponentially with distance from the indenter.

The indenter used in the simulation was 0.060″ in diameter thus only 0.030 inches are shown in the half-sensor cross section. The indenter with the same dimensions as a forcel, i.e. 0.0625″x0.125″, has the closest match with the simulated indenter. The half-width along the narrow axis of the probe is 0.0313″ inches. Figure 3.16 showed the size of the forcels with respect to the mesh used in the simulation. The gross nature of the forcel to mesh element correspondence is such that the difference between the simulated probe and the actual probe of one-eighth of a mesh element width is insignificant. Also note that the simulation is an approximation to the experimental case since the simulated probe was circular in cross section and the actual probe had a rectangular profile. This difference is also insignificant given the rectangular nature and gross distribution of the forcels. The approach taken for comparison was to include only the samples along the axis of highest forcel density. Figure 3.19 illustrates the average of a number of trials using different target forcels and the probe with the same shape and area as a forcel.

The simulation data contained in Figure 3.15 can be viewed in light of the mesh-to-forcel correspondence shown in Figure 3.16. The deflections of all of the elements corresponding to a given forcel can be averaged to provide an estimate of the pressure expected across the area of that forcel. When this is done, the pressure predicted for the central forcel, i.e. the forcel directly under the probe, is estimated to be roughly 250 times greater than the average pressure felt by the next adjacent forcel which, in turn, is 5 times greater than the pressure on the next most distant forcel. The pressure felt by a forcel is transduced into an output reading for that unit.

The rate of decrease seen in Figure 3.19 is considerably less than predicted by the simulation. The differences are due to a number of factors. To begin with, recall that the correction technique used to normalize the output ranges for the forcels used linear interpolation over the range while the actual response was non-linear. In addition, the model of the change in contact resistance due to pressure used to relate the pressure distributions to the sensor output was a simple linear scaling. In reality, contact resistance is a nontrivial phenomenon

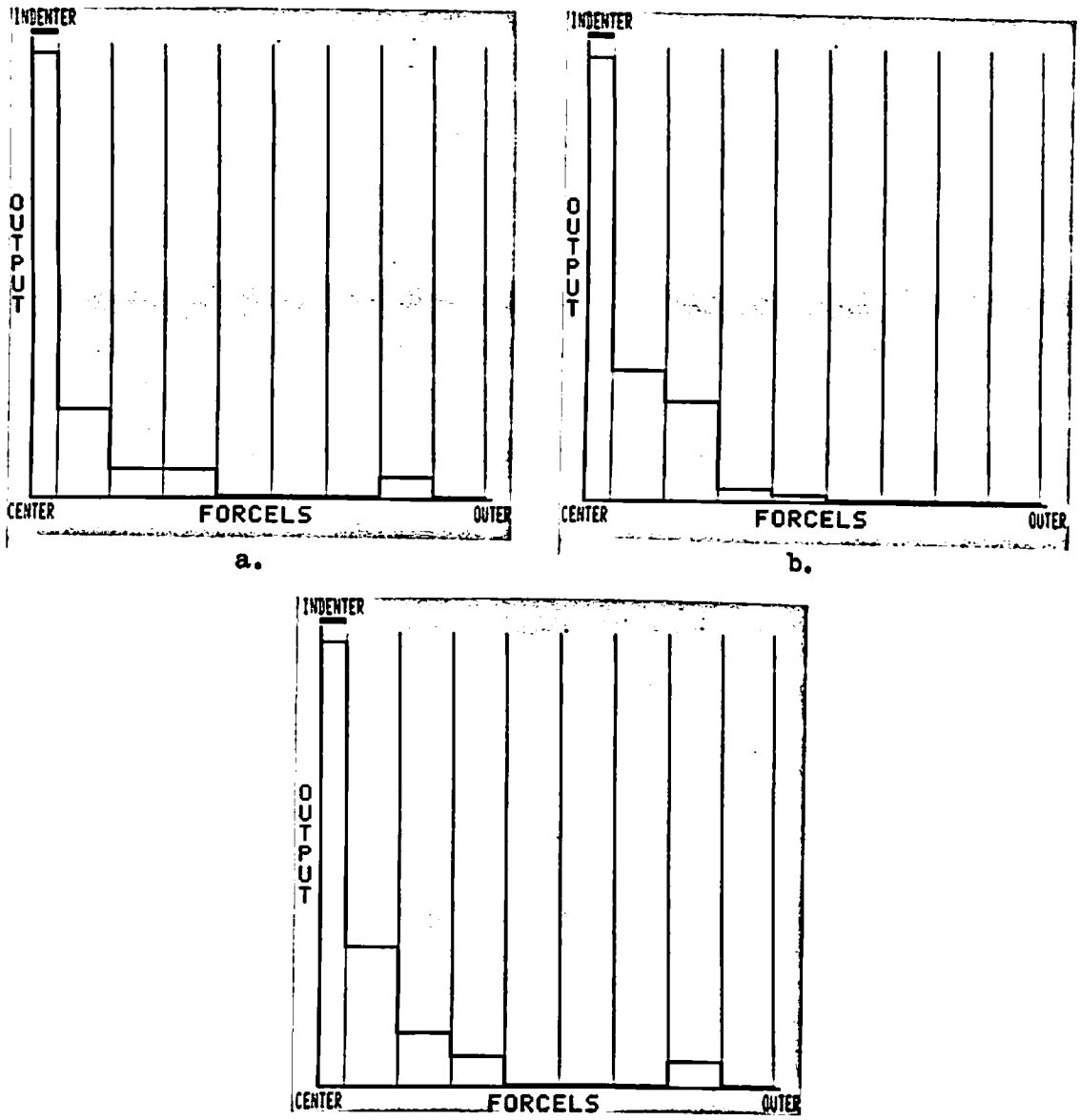


Figure 3.18

Graphs of the data in Figure 3.17. These graphs represent cross sections through the images in 3.17 and are displayed as were the forcels in Figure 3.16. As in Figure 3.17, 3.18a resulted from the smallest indenter, 3.18b from the medium indenter, and 3.18c from the largest.

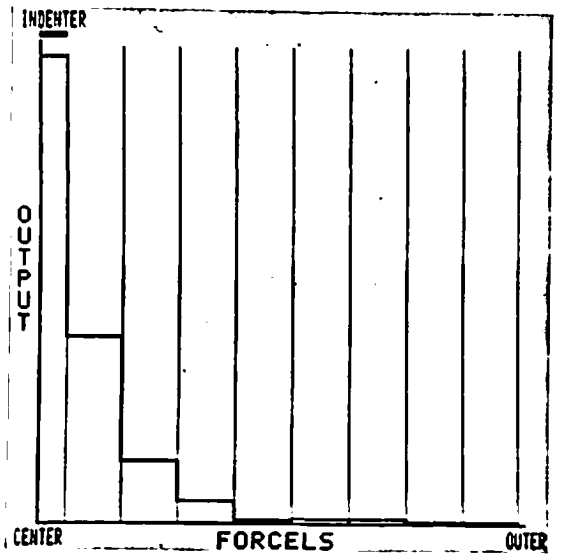


Figure 3.19

This graph contains the average of a number of trials using different target forcels and the probe with the same shape and area as a forcel.

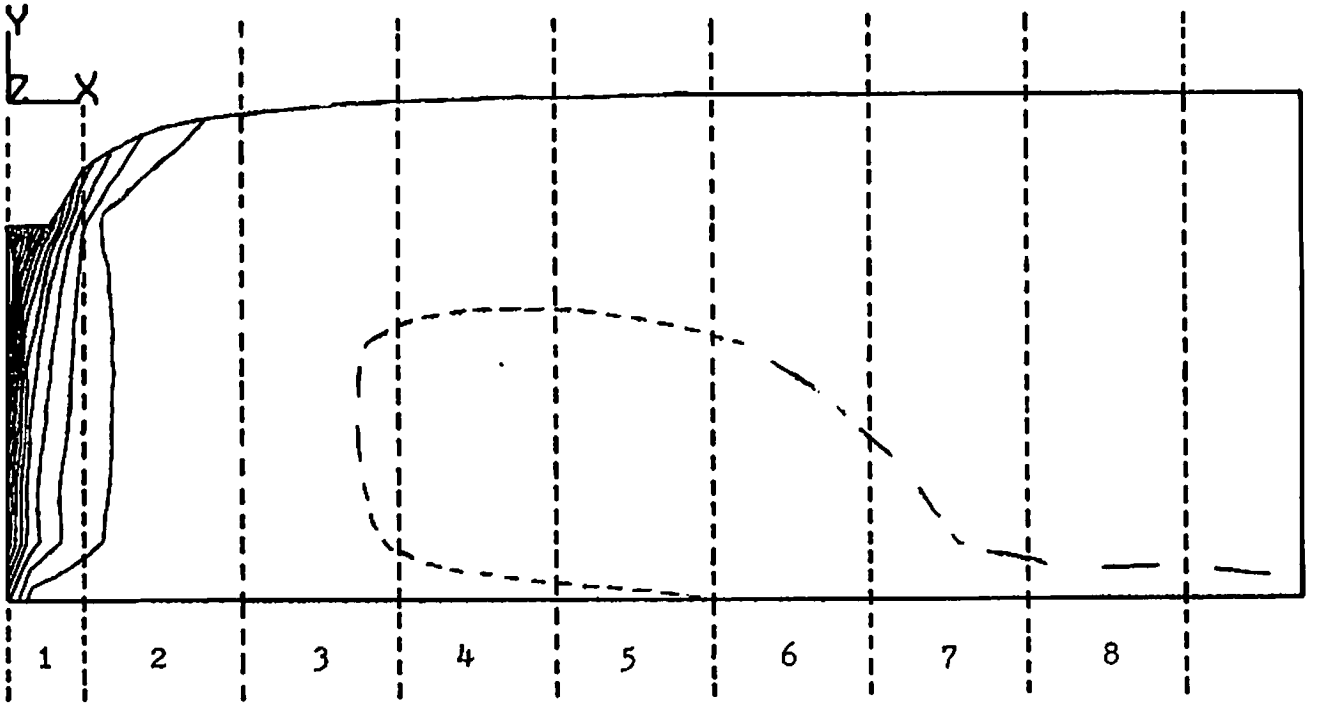
and a linear model cannot completely predict the behavior. Part of the problem here is that the contact was modeled as if it covered the entire area of the forcel where, in fact, it was a point contact located somewhere within the forcel boundary. Finally, the output of the sensor is a function of three variable resistances, only two of which have been considered in this discussion. The resistance of the loop of carbon-filled rubber varies based upon the stress it experiences, and these stresses are not uniform over the volume of the forcel. Figure 3.20 contains a plot of the iso-stress contours in the pad. The non-uniform distribution of contour lines illustrates the changes in stress within a forcel.

A set of average deflections (analogous to the average pressures) can be calculated from Figure 3.15 by averaging the deflections of the elements corresponding to a single forcel. Comparison of this set with the results in Figure 3.19 shows better agreement than comparison with the average pressures. This supports the contention that the resistance of the carbon-filled material is significant compared to the contact resistances.

What the model and simulation have provided is a bounding example of the pressure distribution in a pad similar to the sensor pad used in this device. It showed that in the best case, the pressure distribution due to a sharp edge should decrease by a factor of over 200 between the stimulated forcel and the next adjacent forcel given the dimensions of the forcels in the present sensor.

Figure 3.20

This contains a plot of the iso-stress contours in the pad as predicted by the ANSYS simulation. The forcel boundaries are overlaid on the mesh and the non-uniform distribution of contour lines illustrates the changes in stress within a forcel.


Y
Z
X
1
2
3
4
5
6
7
8

The experimental results showed a decrease over several forcels indicating that a sharp edge is blurred by the sensor which gives it a ramp like appearance several forcels wide. This, of course, implies limits on the ability of the sensor to discriminate between edges with varying degrees of sharpness.

3.3 Calibration Methods

This section discusses the various correction techniques which were explored in an attempt to develop an efficient, automatic procedure for removing the non-linearities and offsets inherent in the sensor data. As was seen from the discussion in the previous chapter, the characteristics of the testing system indicate that the response should be interpreted based upon the deflection of the sensor rather than the applied force.

3.3.1 Introduction

A total of six methods were investigated, including use of the raw data with no preprocessing. The six methods fall into three classes based on their complexity. The simplest class involves doing nothing, i.e. using the data as they are obtained from the sensor. The second class involves subtracting a value determined as the no-force response and followed by linear scaling using a full-force response value. The final two methods fall into a class where the deflection at each forcel is interpolated from the calibration information obtained earlier.

In order to understand how these methods relate to one another, it is useful to examine the extent of the information employed by each. We can plot each of the methods as a point on a graph relating the complexity of the correction algorithm to the extent of the information used. Each method uses information derived from a set of calibration images to normalize each forcel's response curve. In the simplest (non-trivial) algorithm, the information assumes the form of a single value obtained from a single image, while in the most complex algorithm, a data set is built for each forcel from a set of calibration images. We can define abscissa of the graph as representing the type of information used. At one extreme, we have completely global information; at the other, the information required is local, i.e. each forcel has its own calibration curve. The other characteristic of the algorithms of interest is their complexity, since we require that the correction process take as little time as possible. The ordinate of the graph can be used to represent the complexity of the algorithm. Figure 3.21 contains such a graph with a point plotted for each of the algorithms. The succeeding sections provide the details of the methods as well as their advantages and disadvantages. Equations defining each of the correction methods are given in Table 1. Finally, a statistical comparison between the behavior of the various methods is given.

3.3.2 Method 1

The simplest possible correction method involves using the raw data as they are received from the sensor. The data received from the sensor is simply

viewed as an image. The problem with this approach is that the differences in the responses of the forcels is not corrected. In some cases the raw data may be sufficient: however, most algorithms rely on an underlying assumption that all of the transducers (either pixels or forcels) have essentially the same response characteristics. The advantage of this approach is obvious: it requires no computation or additional memory.

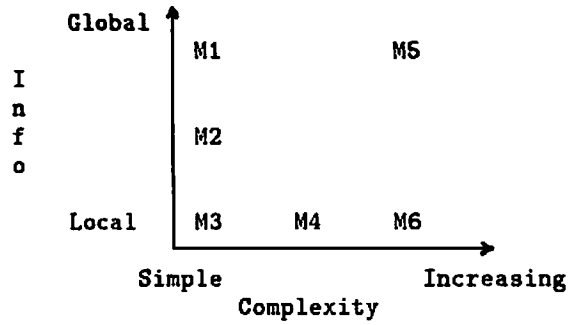


Figure 3.21

3.3.3 Method 2

This is the first method in the second class. It is intended to remove the offset in the response and to scale the data to a specific range. The average value of all of the forcels in a no-force image is subtracted from each of the values in the image to be corrected to produce a modified image. The average no-force value is subtracted from the average value of all of the forcels in a full-force image to determine the response range of the sensor. The single value calculated as the response range is used to scale each of the values in the modified image. This technique uses global information in the form of the averages of the no-force and full-force images.

The strength of this technique lies in the fact that the computation required to correct an image is minimal and only two extra values need to be stored, i.e. the average no-force and full-force values. The complexity of the algorithm is thus very low. An additional benefit of this approach arises from the fact that the correction information employed is calculated from only two images: a full-force and a no-force image. These images can easily be updated whenever the gripper is not holding an object. A no-force image can be loaded whenever the fingertips are not touching and a full-force image is available by simply closing the gripper with the maximum possible force. It should be noted that applying the maximum force over the entire surface of the sensor results in a maximum force value at each individual transducer. This force is less than could be applied to that transducer by applying the same total force but over a smaller area, e.g. the smaller contact area of an object being manipulated. However, such a full-force image can be used to update the model of the response characteristics from moment to moment. The problem is that while each of the

transducers responds in a unique way, they are all treated as if they respond in a uniform way. That is, global information is used to correct totally local errors. In addition, the scaling performed is linear and, as was discussed earlier, the form of the response of forcels is not necessarily linear.

3.3.4 Method 3

This method is a direct extension of Method 2. In this case, the average no-force and full-force values are replaced by complete tactile images. The full-force image was taken from the uniform force experiment utilizing the proboscis and ball-plate device described earlier, cf. Figure 3.5. The no-force image is first subtracted from both the image to be corrected and the full-force image. The latter image then contains the response ranges of the individual forcels. The forcel in the modified image is scaled using the range information for that forcel to produce the corrected image.

This method is only slightly more computationally intensive than the previous method but requires the storage of images in addition to the one to be corrected. As with the previous method, updating of the information used by the calibration method, i.e. the full- and no-force images, can be updated while the sensor is in use. No special calibration experiments or hardware are needed. This method corrects a deficiency in the previous method by using local information, i.e. the images as opposed to single values, to correct local errors. However, the use of the ball and plate to apply a uniform force to the sensor does not, in fact, accomplish that task. The image that is then used as the full-scale image is not truly representative of the maximum sensor output range.

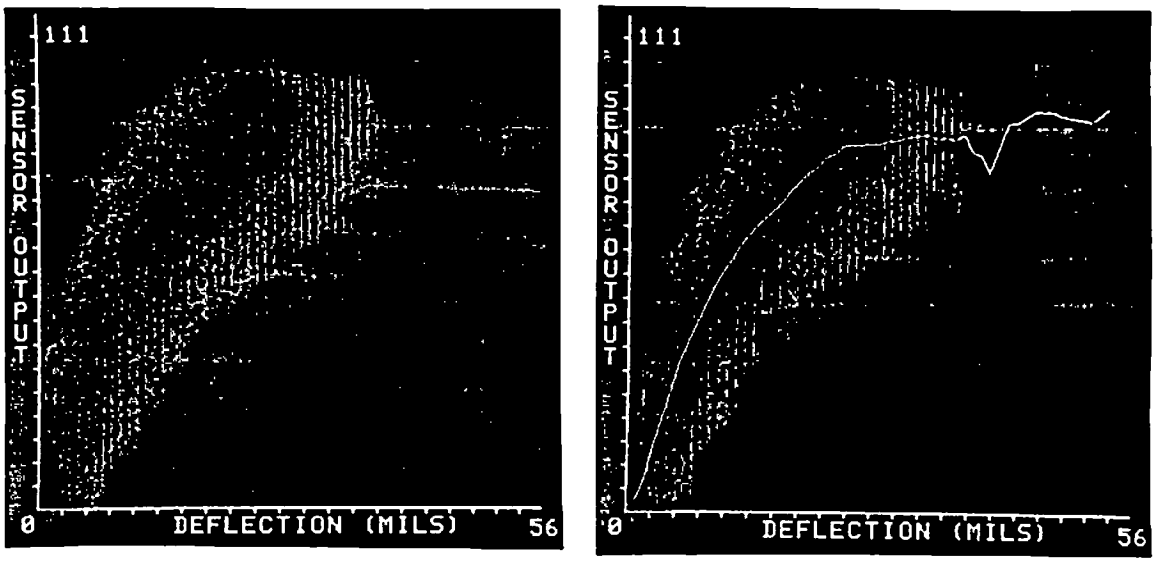
3.3.5 Method 4

To alleviate the previously mentioned problem, an image composed of the maximums obtained during the single-forcel probe experiments was used in place of the full-force image in Method 3. This image differs from the full-force image obtained from the ball-plate experiment in that the data were collected when the force was applied to only the forcel in question. In this case, truly local information is used to correct local errors.

The major problem with this method, as well as with Methods 2 and 3, is that the method assumes that the output of the transducers is linear over the entire response range. That is, the scaling function normalizes each forcel's output into the same range. As can be seen from the previous discussion on the response characteristics, the transducers rarely respond in a linear fashion. A linear approximation to the actual response curve is usually inappropriate. The final two methods interpolate the corrected values directly from the calibration information. An additional problem of this method lies in the fact that the full-force image was obtained from the single-forcel probe experiments. Collecting this information is a tedious task requiring a controlled calibration device. Due to the nature of this procedure, it can only be done rarely; thus, on-line updating of the response behavior is impossible.

3.3.6 Method 5

The idea behind this method is to produce a generic model of the response curves of the transducers in a given sensor. All of the data from the single forcel probe experiment discussed above was combined to produce a single trace. The data points from all of the responding forcels was median filtered to produce the set from which the corrected values are interpolated. A scatter plot of the data is given in Figure 3.22a. A median filter with a window width of 1.5 units was used to produce the curve used for the interpolation. The curve is displayed over the original data in Figure 3.22b.



a. b.

Figure 3.22

3.22a: This graph shows all of the data from the single forcel probe experiment for one sensor plotted on one coordinate system. The data were then median filtered to produce the set from which the corrected values were interpolated in method 5. A window width of 1.5 units was used for the filter. The resulting curve is displayed over the original data in 3.22b.

For each forcel in the image to be corrected, the raw data value is used as coordinate along the abscissa of the graph in Figure 3.22b. The "corrected" value is determined as the coordinate along the ordinate of the graph of the point on the median filtered curve whose abscissa value equals the raw data value. Where the filtered data set does not contain the exact abscissa value of the raw sensor value, linear interpolation is used to calculate the deflection value from the deflection values corresponding to the closest pair of sensor values.

This method provides a better fit to the data as collected during the

response characterization experiments but uses the same data set for each of the forcels, i.e. the data set calculated by median filtering the set composed of all of the response pairs from all of the responding transducers. In this sense, the information employed is global in nature. While these data more accurately reflect the actual response curves, tactile images corrected by this method lose all definition. We feel that this behavior is due to the affects of the filtering phase. Compared with the previous algorithms, this method is relatively complex. Not only is the interpolation phase more expensive, computationally, than the previous techniques, but a great deal of data must be collected under controlled conditions in order to build the response curve data set.

3.3.7 Method 6

The final method uses the interpolation strategy of Method 5 but uses only the data for the transducer being corrected. That is, for each forcel, the responses obtained for that forcel during the single forcel probe experiment are median filtered and the resulting set used for the interpolation phase. The interpolation phase is identical to that used in the previous method except that each forcel now has its own set of response data. This method has the same computational requirements as the previous method yet requires a great deal more memory since there is now a data set for every forcel. Table 1 contains the updating expressions for the six correction methods.

3.3.8 Comparison of the Methods

Once the six correction methods had been defined, a means for comparing their behavior was needed. Two levels of comparison were undertaken. The set of images collected during the uniform force experiments was used for the first comparison.

The purpose of the correction procedure is to compensate for differences in the response characteristics of the forcels. That is, a given stimulus applied to different forcels should produce the same numeric value in the corrected image. We recall that the uniform force experiments were designed to produce a set of images where force was applied equally to all forcels. The force was ramped from zero up to a maximum value and then down to produce an entire set of images. We can then apply each of the correction methods to each of the images in the set. If a correction method works properly, the corrected images should be flat, i.e. all of the values in the image should be equal, since the same stimulus was applied to each forcel. This "flatness" can be measured as the standard deviation of the values in the image. In addition, the graph of the mean values of the images should increase and then decrease in value with the same shape as the original force ramp.

The set of 61 uniform force images collected using the ball and plate apparatus was used to compare the methods. For each image in the set, all six correction methods were applied and statistics collected across the resulting images. In addition, each of the corrected images was averaged using a 3x3 forcel window; then statistics were again collected.

Method	Forcel Update Equation
M1	$C\_Image(j,k) = Raw\_image(j,k)$
M2	$C\_Image(j,k) = \frac{(Raw\_image(j,k) - \overline{No\_force\_image})}{\overline{No\_force\_image} - \overline{Full\_force\_image}}$
M3	$C\_Image(j,k) = \frac{Raw\_image(j,k) - No\_force\_image(j,k)}{No\_force\_image(j,k) - Full\_force\_image(j,k)}$
M4	$C\_Image(j,k) = \frac{Raw\_image(j,k) - No\_force\_image(j,k)}{No\_force\_image(j,k) - Ind\_FF\_image(j,k)}$
M5	$C\_Image(j,k) = Interpolate(Data\_set, Raw\_image, j, k)$
M6	$C\_Image(j,k) = Interpolate(Data\_set(j,k), Raw\_image, j, k)$

Table 1

This table contains the updating equations for the six correction methods. C_image, Raw_image, Full-_force_image, No_force_image, and Ind_FF_image denote the corrected, raw, max- imum applied force, zero force, and full force data from the individual probing experiments images, respectively. A bar is used to denote the average of a data set. The (j,k) term indicates that a single forcel is addressed. Data_set refers to the median filtered data collected during the single forcel probing experiments. Finally, "Interpolate" represents the interpolation process as defined in the discussion of Method 5.

Figure 3.23

This figure diagrams the tactile image correction comparison scheme. A set of 61 uniform force images collected using the ball and plate apparatus. All six correction methods were applied and statistics collected across the resulting images. A second set of images was created by applying a 3x3 window average to each of the corrected images and statistics again collected. The statistics were collected over two regions: the entire image and the image less the outermost row and column. Seven measures were calculated for each image including: the mean, standard deviation, and variance of the values, the mean percentage difference of each forcel from the mean, the mean of the squares of the percentage difference of each forcel from the mean, the mean absolute value of the vertical and horizontal neighbor differences, and finally, the mean of the square of the neighbor differences.

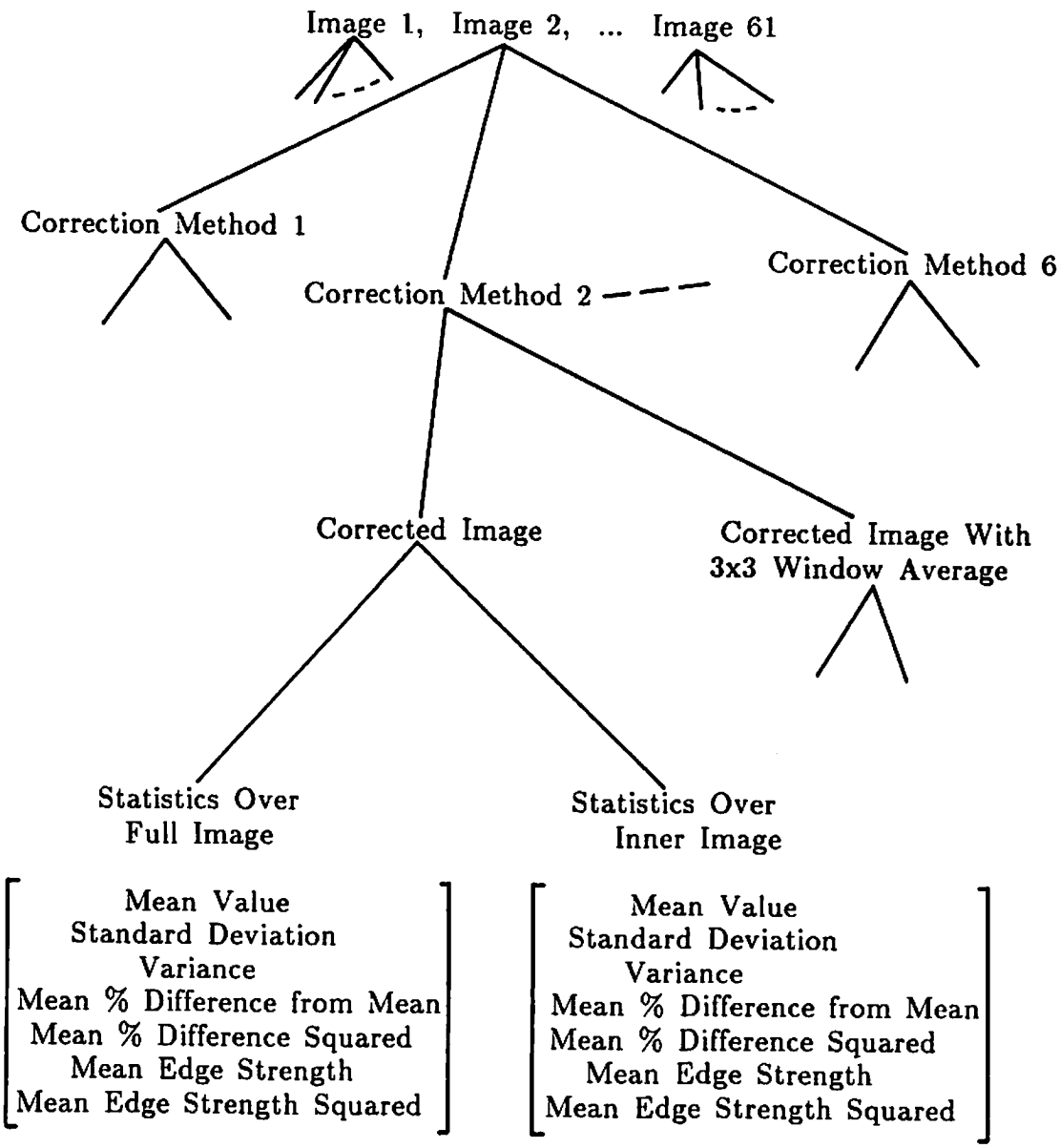


Image 1, Image 2, ... Image 61
Correction Method 1
Correction Method 2
Correction Method 6
Corrected Image
Corrected Image With 3x3 Window Average
Statistics Over Full Image
Statistics Over Inner Image
Mean Value
Standard Deviation
Variance
Mean % Difference from Mean
Mean % Difference Squared
Mean Edge Strength
Mean Edge Strength Squared
Mean Value
Standard Deviation
Variance
Mean % Difference from Mean
Mean % Difference Squared
Mean Edge Strength
Mean Edge Strength Squared

The statistics were collected over two regions. In the first case, all of the forcels in the image were included. In the second case, those forcels in the outermost rows and columns were not included. This second case was intended to reduce the affect of bad forcels located along the periphery. A total of seven measures were calculated for each image. These included: the mean, standard deviation, and variance of the values, the mean percentage difference of each forcel from the mean, the mean of the squares of the percentage difference of each forcel from the mean, the mean absolute value of the vertical and horizontal neighbor differences, and finally, the mean of the square of the neighbor differences. Figure 3.23 graphically illustrates the comparison data generation scheme.

Table 2 contains the mean values for each of the statistics for the six correction methods. The values are scaled so that each image has the same mean. In addition, the statistics are shown for both the entire image and the image less the edge forcels. Table 2a contains the statistical values collected over the entire area of each image. Table 2b contains the values resulting when the perimeter forcels are not considered in the statistics and Table 2c shows the statistical values from Table 2b in terms of the percentage of the minimum value. Thus a statistical value equal to the mean appears as 100%. Table 3 contains the same information as Table 2 however a single pass of 3x3 window averaging was applied to each image prior to collecting the data.

As stated earlier, if a correction method is performing perfectly, both the standard deviation of the values in a corrected uniform force image should be zero and the set of means should increase and then decrease in the same fashion as the force at the time of collection of the images. Figure 3.24 contains the standard deviation graphs for the six methods.

Recall that since the stimulus was uniform over the entire image in each case, the "corrected" image should have the same value for all of the forcels. Thus the standard deviation of each image should be zero. Figure 3.24a and b contain the graphs of the data after correction yet without local averaging. Figure 3.24a includes all forcels while 3.24b contains only the inner forcels, i.e. not including the boundary elements. The edge effects are very clear in the graph of method 4. Notice that when the boundary elements are not considered, the graph is much smoother. This shows that the correction method does not work effectively for the forcels along the edge of the pad. Therefore, the response of these elements cannot be modeled in the simple manner employed by the correction scheme. Figures 3.24c and d contain the standard deviation graphs obtained after a 3x3 window average is applied to the corrected data. As expected, not only are the magnitudes of the standard deviation reduced but the graphs show smaller variations. Figure 3.24d was derived from the data excluding the boundary forcels. From this graph, correction method 4 appears to have the best behavior in the maximum force range while method 6 appears best in the lower force range. However, method 6, while faring well during evaluation using this metric, performs poorly in other situations, see below.

Table 2

This table contains the means values for each of the statistics for the six correction methods collected over both the entire image over the image less the outer rows and columns. The values have been scaled so that the images have the same means. Table 2a contains the statistic values collected over the entire area of each image. Table 2b contains the values resulting when the perimeter forcels are not considered in the statistics. Table 2c shows the statistic values from Table 2b in terms of the percentage of the minimum value. Thus a statistic value equal to the mean appears as 100%.

Correction Method	Image Measure						
	Image Mean	S.D.	Var.	% delta	Sq. % delta	Edge	Sq. Edge
1	99.58	65.26	4467	91.41	10914	59.00	8313
2	99.58	72.17	5587	79.38	8278	90.18	21795
3	99.58	51.54	3071	38.35	3665	59.72	30590
4	99.58	51.81	2963	34.73	1759	93.40	41391
5	99.58	56.97	3351	23.75	893	167.89	47905
6	99.58	42.69	1966	10.31	123	214.73	75496

Table 2a

Correction Method	Image Measure						
	Image Mean	S.D.	Var.	% delta	Sq. % delta	Edge	Sq. Edge
1	100.88	59.26	3872	89.19	10837	54.40	7872
2	100.88	67.21	5044	80.90	9015	85.07	42116
3	100.88	29.54	935	24.80	965	49.88	16228
4	100.88	38.42	1604	27.78	1049	77.81	30630
5	100.88	49.56	2611	22.15	828	136.86	33432
6	100.88	33.31	1210	9.69	107	157.29	38496

Table 2b

Correction Method	Image Measures as % of Minimum						
	Image Mean	S.D.	Var.	% delta	Sq. % delta	Edge	Sq. Edge
1	100	200	414	920	10042	109	100
2	100	227	539	834	8354	170	535
3	100	100	100	255	894	100	206
4	100	130	171	286	972	155	389
5	100	167	279	228	767	274	425
6	100	112	129	100	100	315	489

Table 2c

Correction Method	Image Measure						
	Image Mean	S.D.	Var.	% delta	Sq. % delta	Edge	Sq. Edge
1	99.78	37.71	1468	31.73	888	37.46	3480
2	99.78	40.24	1724	28.64	730	56.49	8238
3	99.78	27.87	860	16.05	354	40.03	9977
4	99.78	31.87	1111	14.19	188	63.00	14107
5	99.78	32.20	1063	9.28	77	91.97	16433
6	99.78	25.39	688	3.54	11	125.79	28974

Table 3a

Correction Method	Image Measure						
	Image Mean	S.D.	Var.	% delta	Sq. % delta	Edge	Sq. Edge
1	101.15	30.18	957	32.82	939	34.05	3511
2	101.15	33.36	1190	29.40	770	51.09	9937
3	101.15	19.36	414	12.54	231	36.39	10678
4	101.15	23.00	565	12.51	135	54.11	13417
5	101.15	24.75	638	8.87	70	70.39	9217
6	101.15	18.30	364	3.28	9	85.14	13091

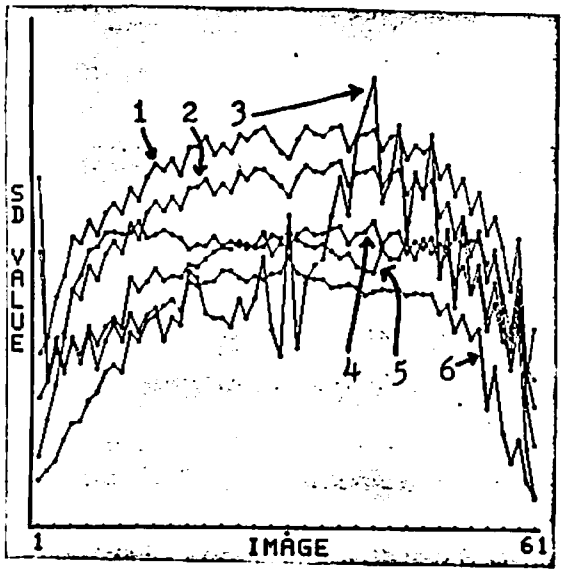
Table 3b

Correction Method	Image Measures as % of Minimum						
	Image Mean	S.D.	Var.	% delta	Sq. % delta	Edge	Sq. Edge
1	100	164	262	1001	10039	100	100
2	100	182	326	896	8227	150	283
3	100	105	113	382	2476	106	304
4	100	125	155	381	1447	158	382
5	100	135	175	270	757	206	263
6	100	100	100	100	100	250	373

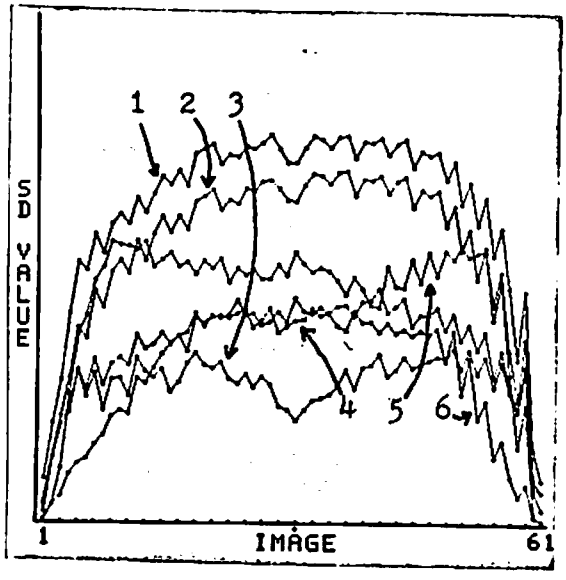
Table 3c

Table 3

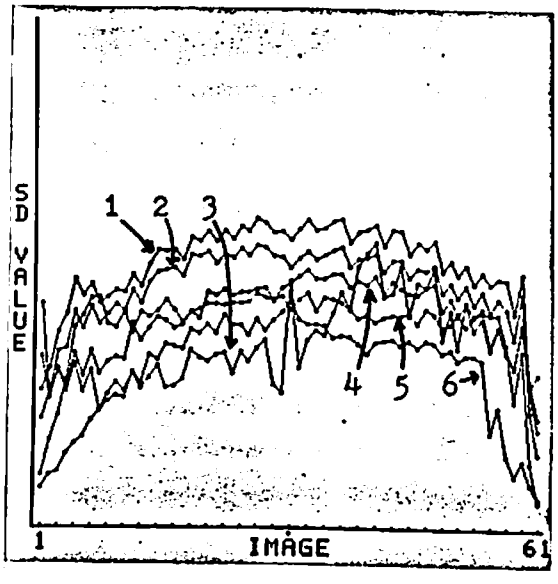
This table contains the same information as Table 2 but a single pass of 3x3 window averaging was applied to the images prior to collecting the statistics. This has the affect of smoothing the images and thus reducing some of the statistic values. Table 3a contains the statistics for the full images, 3b for the images less the perimeter forcels, and 3c contains the data in 3b in terms of percentages of the minimum values.


SD VALUE
1 2 3
4 5 6
1
IMAGE
61

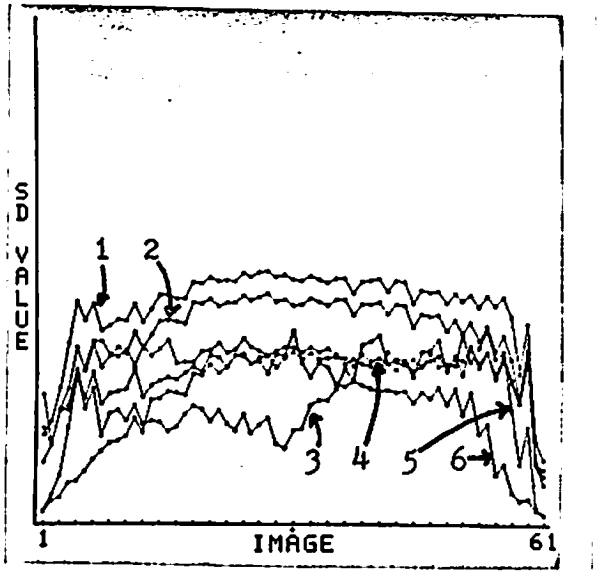
a.


SD VALUE
1 2 3
4 5 6
1
IMAGE
61

b.


SD VALUE
1 2 3
4 5 6
1
IMAGE
61

c.



d.

Figure 3.24

These graphs show the standard deviation for the six correction methods. 3.24a: The statistics for these graphs were collected over the entire image area while the graphs in 3.24b represent data collected over the inner images only. 3.24c again contains data derived from all of the forcels but a single pass of a 3x3 window average has been used to smooth the data. Similarly, 3.24d contains data from the images less the outer forcels after a single pass of local averaging has been applied. The set of images provides the coordinate system for the abscissa while the range of the standard deviation is measured along the ordinate. The numeric labels on the curves correspond to the correction method underlying the data.

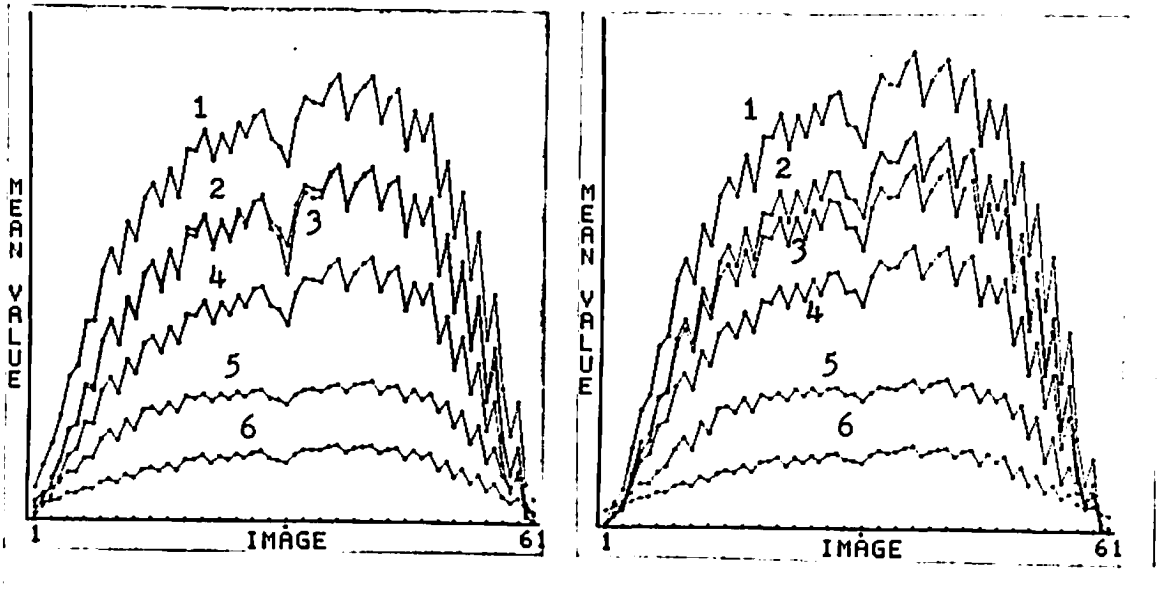
Throughout the remainder of this work, method 4 will be used whenever correction is performed on an image.

Figure 3.25 contains the graphs of the mean values of the images. Each data point along the abscissa represents the average of an image from the set of 61. The images range from left to right going from no applied force to the maximum and back to the minimum by the right end of the abscissa. The peak applied force thus occurs in the image in the middle of the axis. Recall that the A/D conversion system introduced fluctuations in the reported force values from the fingertips. These fluctuations are seen as the jaggedness of the curves. Figure 3.25a contains the average calculated over the entire image. The results are as one would expect in that methods 3 and 4 produce very similar results. It is interesting to note that the most complicated method, method 6, tends to suppress the range of the data. The results of this are shown below. Figure 3.25b contains the graphs of the average taken over all forcels except those along the edge of the image. The boundary forces tend to show the most variation in response, probably due to the mechanical nature of the sensor and the edge effects this introduces. 3.25c contains data derived from all of the forcels but a single pass of a 3x3 window average has been used for smoothing. Similarly, 3.25d contains data from the images less the outer forcels after a single pass of local averaging has been applied.

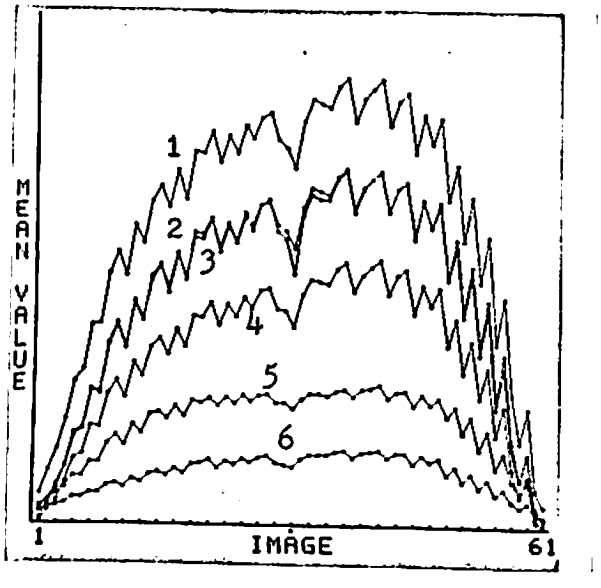
This evaluation technique used uniformly applied force. It is rarely the case in reality that a force image of interest will be the result of a uniformly applied force. Instead, it is the variations in the applied force that are of interest. It is necessary, then, to compare the correction methods in real situations. Unfortunately, evaluation of these results can be accomplished only on a subjective basis. Figure 3.26 contains the results of the application of the correction methods, both with and without local averaging, to a typical tactile image.

The data image, Figure 3.26a, was produced by a cylindrical steel shaft with its major axis inclined 30 degrees from the vertical axis. Each image in the left column in the Figure contains the raw data image corrected by one of the algorithms: Figure 3.26c corresponds to correction by method 2, 3.26e by method 3, etc. The right hand column contains the image after a single pass of 3x3 local averaging on the corresponding corrected image. Thus Figure 3.26h contains the local averaged version of the data corrected by method 4, Figure 3.26g.

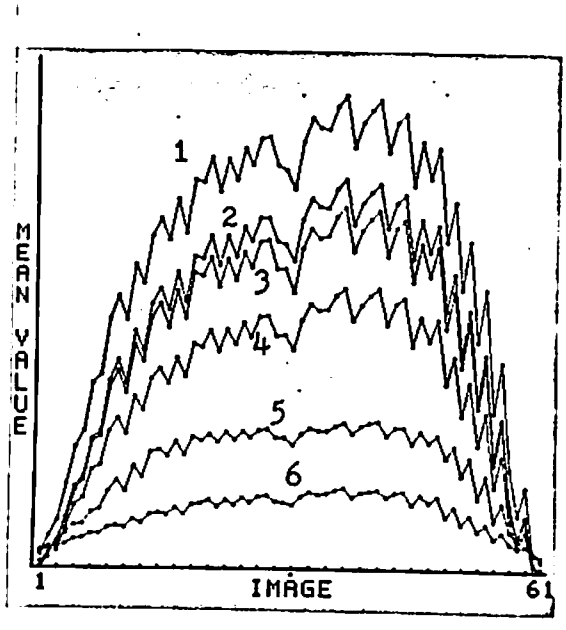
There are two interesting things to note. While all of the algorithms -- with the exception of method 6 -- result in images which still contain the elongated structure seen in the raw data, only methods 3 and 4 remove the boundary effects seen along the upper edge of the image. Methods 5 and 6, while being the most complicated, result in images with the poorest appearance. Method 6, in particular, seems to remove definition in the image. A subjective rating of the correction capabilities places methods 3 and 4 at the top of the list with method 6 at the bottom.


MEAN VALUE
1
2
3
4
5
6
1
IMAGE
61
MEAN VALUE
1
2
3
4
5
6
1
IMAGE
61

a. b.


MEAN VALUE
1
2
3
4
5
6
1
IMAGE
61

c.



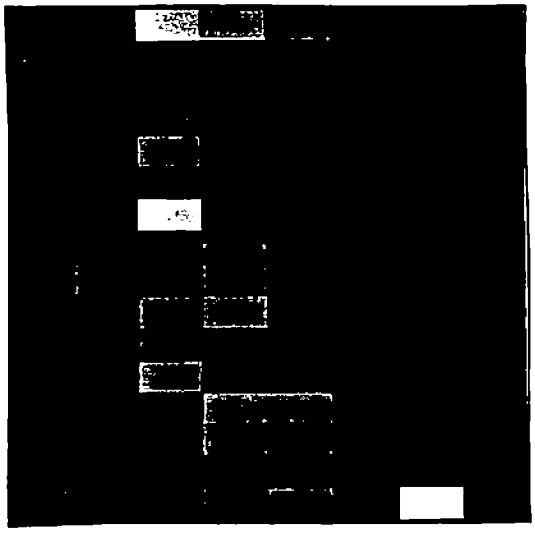
d.

Figure 3.25

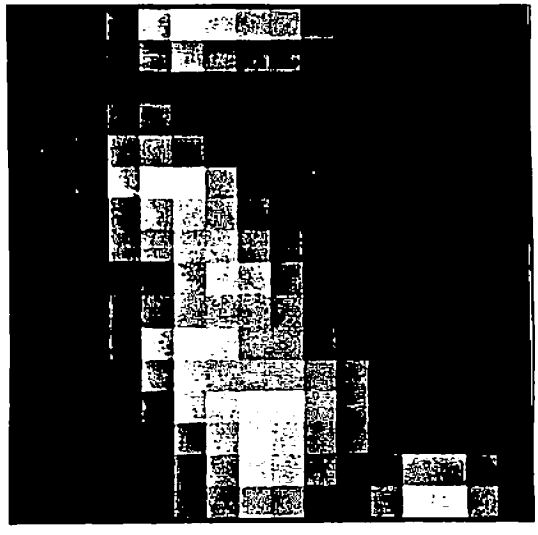
This figure contains graphs of the mean values of the images. Each data point along the abscissa represents the average of an image from the set of 61. The peak applied force occurs in the image in the middle of the axis. 3.25a contains the average calculated over the entire image. 3.25b contains the graphs of the average taken over all forcels except those along the edge of the image. 3.24c again contains data derived from all of the forcels but a single pass of a 3x3 window average has been used to smooth the data. Similarly, 3.24d contains data from the images less the outer forcels after a single pass of local averaging has been applied.

Figure 3.26

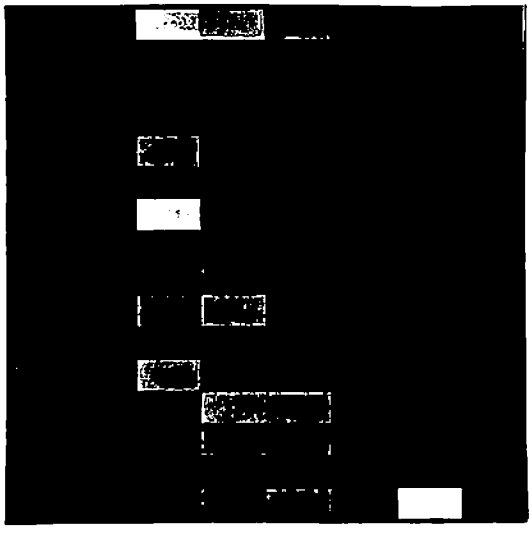
These twelve images contain the results of applying the correction methods to the same tactile image. The raw data image appears in 3.26a and was produced by a cylindrical steel shaft inclined 30 degrees from vertical. The images are grouped in pairs with the right image in each pair derived from the left image by the application of a single pass of 3x3 window local averaging. 3.26a: M1. 3.26b: M1 plus 3x3 averaging. 3.26c: M2. 3.26d: M2 plus 3x3 averaging. 3.26e: M3. 3.26f: M3 plus 3x3 averaging. 3.26g: M4. 3.26h: M4 plus 3x3 averaging. 3.26i: M5. 3.26j: M5 plus 3x3 averaging. 3.26j: M6. 3.26k: M6 plus 3x3 averaging.



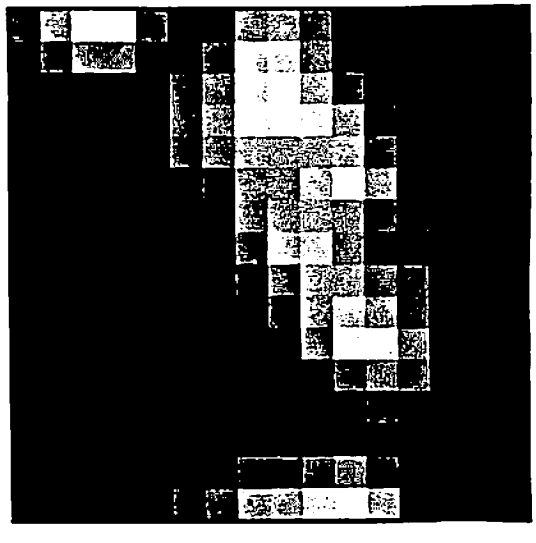
a.



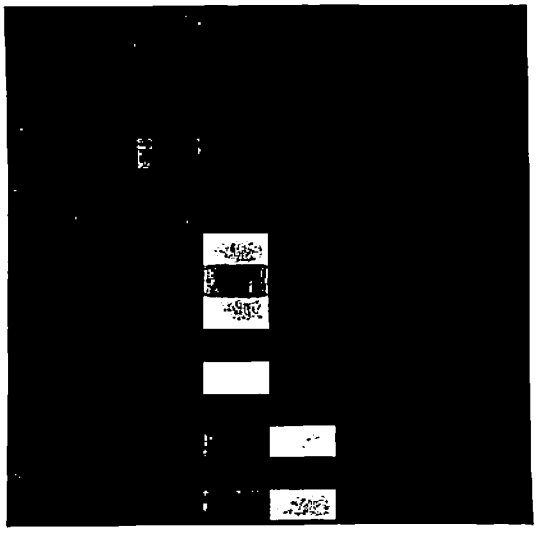
b.



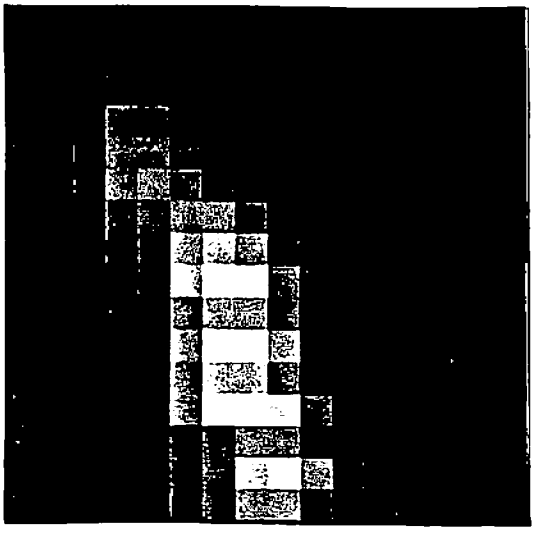
c.



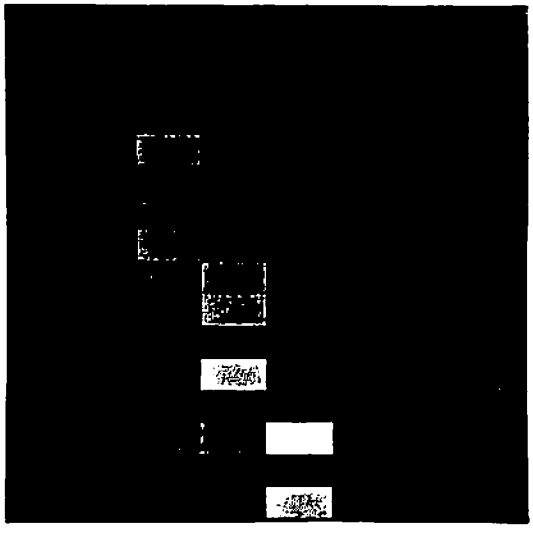
d.



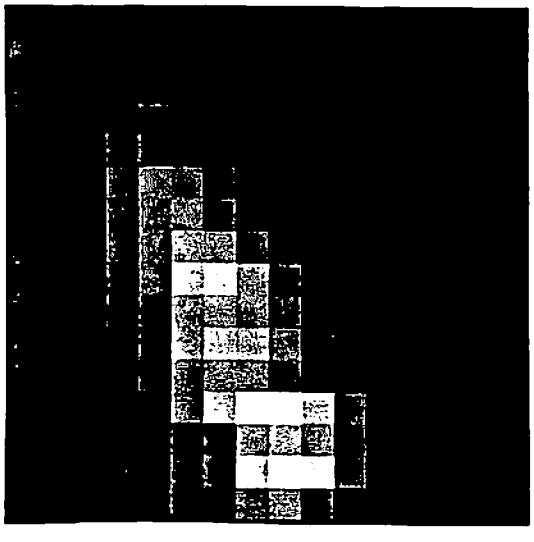
e.



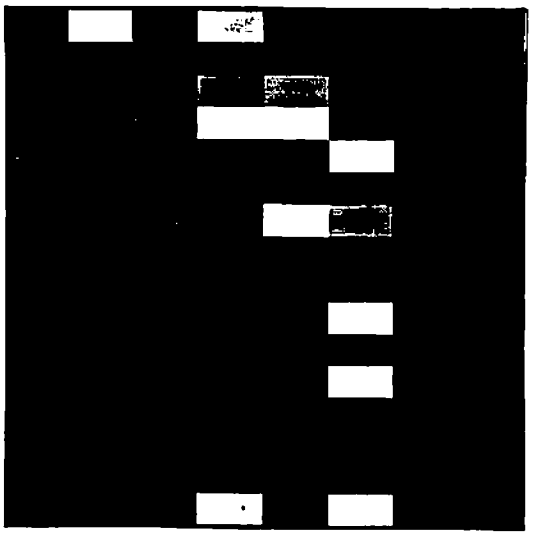
f.



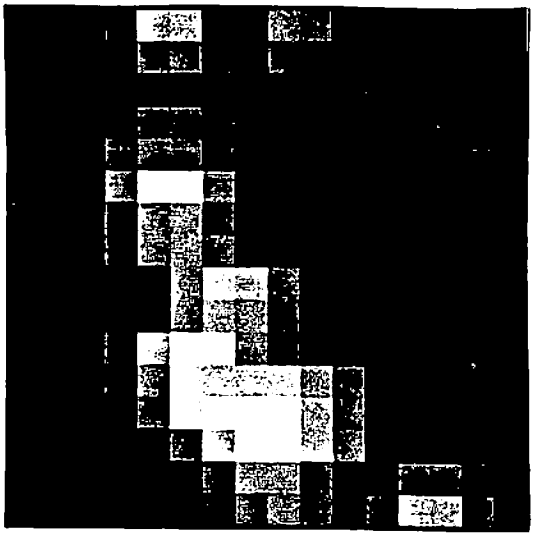
g.



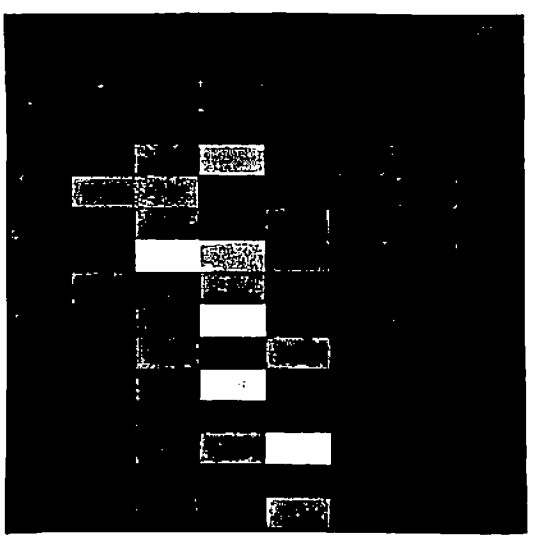
h.



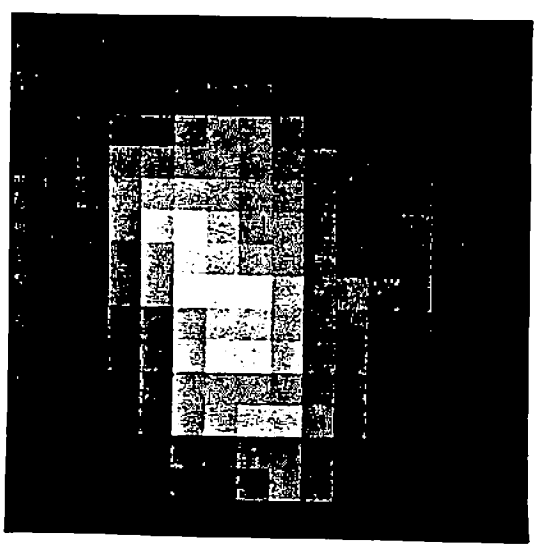
i.



j.



k.



l.

From the uniform force test and the subjective ranking test, it appears that method 4 is a reasonable choice as correction algorithm. As indicated above, method 4 with one pass of a local averaging process is used throughout this work in situations where the image is corrected. It should be emphasized that the point of this correction is not to produce an image wherein the values are indicative of absolute forces or deflections but rather an image in which a particular value found at two forcels implies that the stimuli associated with those forcels were equal.

This chapter has explored the response characteristics of the device developed in Chapter 2. In addition, a series of "correction" algorithms was presented and one of the series was chosen for application in further work. The intent of the correction process is to remove non-linearities and offsets from the raw sensor data so that the output of a given forcel can be compared with that from another forcel. The process used is relatively simple computationally but does not accurately model the exact response form of the transducers. The intent is not to calibrate a forcel's output to some absolute scale but rather to remove dimensions from consideration entirely and deal with the differences in the output across the sensor.

The next logical question concerns how the information contained in tactile images can be extracted and used to control a machine. Chapter 4 addresses issues involved in static tactile image processing, TIP, while issues in dynamic TIP are discussed in Chapter 5.

Chapter 4

Static Tactile Image Processing

4.1 Introduction

Having addressed the topics of robot senses, robot sensors, and tactile sensors and having developed and characterized a tactile array sensor, it is time to address the issues surrounding tactile image processing. Tactile images contain information about the shape of the object sensed, as deflections, as well as the distribution of forces present at the interface with the object. Chapter 3 ended with the development of a "correction" algorithm which allows one to produce images in which a given output value observed at two different forcels can be assumed to have resulted from approximately the same stimulus. The interest here focuses on the processing steps employed to extract useful information from the corrected images.

This chapter begins with a discussion of other relevant work in the field, continues with a discussion of the constraints which may be employed in the tactile domain, and then presents the results of several algorithms used to extract information for control. The chapter ends with a discussion of the logical extensions of the ideas.

4.2 Review of Other Work

The work in tactile sensing has fallen into two general catagories: object identification from grasping and image processing. The work presented in subsequent chapters falls into the second class and thus this class will receive greater attention. Identification from grasping is typified by attempting to understand the contour or shape of an object from an instrumented gripper. In the simplest case, the set of angles for the joints in the fingers of the robot gripper when closed on an object are used as keys to the identity of the grasped object [153] [154]. Gaston investigated the case of the recognition of planar curves from multiple contact points [70]. This senario can be extended to include simple binary touch sensing on the inner surfaces of the fingers. The information in this case includes not only the joint angles but also the pattern of contact [129]. One further extension involves placing a tactile array sensor either in the palm of the hand or several on the fingers [158] and using both the limited shape information from the tactile sensors and the finger positions. This is also the approach of the research team at the University of Pennsylvania [2].

A conceptually similar approach involves utilizing a sensitive probe to search the workspace of the robot. The probe, or finger, used by the University of Pennsylvania group has an octagonal cross section, roughly 3/4 inches in diameter, and a length of approximately 5 inches. Each of the eight major faces contains 16 sensitive elements. The tip of the device is constructed from eight triangles joined together to form a pyramidal structure and is affixed to one end of the basic probe. The apex of the pyramid contains one sensitive element as does every other side. This results in a total of 133 sensitive points on the

probe. The probe is used to sense contact with objects and the kinesthetic information at the points of contact used to construct models of the encountered object and to direct further movement [16].

Wolfeld [217] discusses some preliminary work conducted in regard to the extraction of useful information from a tactile sensing array. The data supplied by the sensor was essentially uncalibrated except to the extent that the values were all scaled to be in the same dynamic range. His first processing technique was designed to determine the angle between the plane of the sensor pad and the plane of an object in partial contact with the sensor. The average of the differences between adjacent forcels was used as an indication of the angle of contact. The calculated results showed excellent agreement with the actual values. This form of information would prove useful for aligning the fingertips with flat faces on an object being grasped.

The approach to dynamic tactile image processing taken by Wolfeld was an attempt to extract information regarding the movement of the sensor relative to the environment [217]. His technique consisted basically of differencing successive images. The zero crossings in the set of difference values at each pixel over time was calculated. The ratio of the number of zero crossings to the size of the number of images considered was used as a feature. By fixing the window size and centering a window at each time step, this ratio was plotted against time. Cross correlation was then used to try to match the curves of nearby transducers, with little success. The purpose of the technique was to determine an operator which could be used to locate similarities in the time course of changes in adjacent transducers.

A different approach to the identification of objects is used by Sato et al. [185]. Recall that the sensor produces a set of binary images representing a contour map of the object sensed where the contours indicated height. These images are processed to extract the contours and a graph structure for the object is constructed which represents the relationships between the contours at various levels. Graph theoretic techniques are used to match a sensed object's graph with those of previously experienced objects in order to determine identity.

A probabilistic approach to object recognition from touch was used by Briot [34]. The input tactile images were standardized so that their barycenters and principal axes were in a specific location and orientation respectively. During a learning phase, each part was presented in each of its stable positions. A probability mask was produced and stored for each position. During the recognition phase, a new tactile image would be recorded, standardized to the desired location and orientation, the probabilities computed for the various objects/positions, and a decision made as to the identification. Briot reports reasonable success with this technique but the disadvantage of mask size prompted him to develop another technique.

This second approach utilized the structure of the tactile images in a more explicit manner. In this case, the set of images of the objects to be identified was analyzed to determine a set of distinguishing features. (Recall that this approach is very similar to that used by most binary vision systems.) The

decision rule applied during the recognition phase is based on likelihood that a sensed tactile image, and its associated parameter values, matches one of the recorded states. Briot indicates that this approach is very successful, as should be expected from the success of the industrial vision systems of similar design.

The final technique involves the use of two "artificial skin" sensors mounted on the fingertips of a robot gripper. The intent was to produce an algorithm capable of identifying objects held in a robot gripper while the object is being handled. The information available to the recognition routine includes two tactile image, one from each fingertip, and the distance between the fingertips. The first scheme mentioned above is applied on this increased amount of information, i.e. joint probabilities are computed and the identity of the object with the greatest probability is chosen. This assembly was mounted on a manipulator which was used to transport parts from one site in the work space to a set of containers. During the transportation phase, the identity of the object was determined and this information was used to choose the path to the appropriate container for the part.

The approach taken by Hillis [98] to object recognition involves extracting certain global parameters describing the tactile image and the stability of the object. The parameters used include shape, i.e. LONG vs ROUND and the existence of bumps, i.e. depressions or small protuberances. The image processing section involves finding the center of mass of the above threshold section of the image. It should be noted that three thresholds are used resulting in four possible regions corresponding to: background, valleys, object surface, and bumps. The major and minor axes are determined and the bounding box aligned with these axes calculated. The shape is determined by the ratio of the dimensions of the bounding box and the bump information from the second and fourth regions in the image. The stability of the object is defined as the ability to roll the object under the finger. A learning phase provides the system with its initial information. The feature vector obtained during an identification test is compared with the vectors from previous experience and a decision made based upon the degree of match.

4.3 Constraints in the Tactile Domain

We have discussed tactile sensors and the work of others in the area of tactile image processing. The lack of major work in this area has led to rather haphazard approaches to the processing of this form of data. At this point, a discussion of the constraints and assumptions applicable in the tactile domain is appropriate, see also [156]. Once these have been outlined, it will be possible to develop techniques for extracting the needed information in an organized fashion. Since most of the image processing work to date has occurred in the visual domain, it is useful to contrast the properties of that domain with those of the tactile domain.

Vision, as characterized by digitized images from TV cameras, is inherently two-dimensional. Thus the third dimension must be inferred from the two-dimensional image. The force/deflection images collected from a tactile array

sensor such as that developed here are inherently three-dimensional. Since the sensor is a contact device, the images produced represent three-dimensional characteristics with respect to the sensor surface. In a robotics environment, the sensors can be utilized in essentially two ways. In one case, the tactile sensors are placed on the fingertips of the robot's gripper. In this case, position and orientation of the sensor is controlled by the manipulator. The other method employs the sensors as passive devices fixed to the workspace. In both cases, the position and orientation of the sensor is known and thus the position and orientation of the tactile images are also known. We can thus see that the problem of recovering the three-dimensional information from the images does not exist.

In addition, vision is typically a passive sense and as such must rely on an energy source located in the environment, i.e. a light source. In the most constrained situations the light is projected by a system controlled by the camera system. In general, however, the location and sometimes number of light sources is unknown. The fact that vision usually requires reflected energy adds several difficulties to the processing of the images. Specularity is the phenomenon in which bright spots appear in the images as the result of direct reflection from the light source. Shadows are also present in image with non-uniform lighting. The processing required to understand which regions are shadows and which are objects is also difficult. In the tactile domain, specularity and shadows do not exist in any form, thus the processing is more straightforward.

The contact nature adds an additional constraint on the image processing; that is, the objects in the image can easily be separated from the background. Since the background of a tactile image represents those regions in which no force is being applied, the separation between an object which applies forces and the background is straightforward. The contact nature of the device does imply, however, that the "view" of the environment is local in nature. That is, the tactile images contain information only about those stimuli which are in contact with the device.

The characteristics of scaling and perspective distortion appear in the visual domain due to the non-contact nature of the sensor. Because the tactile sensors can transduce information only about objects in direct contact, there are no scaling and perspective distortions in the images. Again the contact nature of the sensor adds an additional characteristic. Force must be applied to a stimulus object by the sensor in order to form an image. If the object is non-rigid, this force can deform the shape. Such deformation can lead to difficulties in identifying the object from touch and can introduce permanent, undesired changes in the object.

Occlusion in visual image processing is the phenomenon wherein the image of one object covers or blocks the image of another. Since vision provides a view of distant as well as close objects, the objects can have an arbitrary three-dimensional relationship. Occlusion also exists in the tactile domain yet assumes a different form. Tactile occlusion occurs when two objects are in contact with the sensor and one of the objects is also superimposed between the sensor

and some part of the other object. The contact nature of the sensor provides accurate information about the part of the occluded object being touched. This information and the shape information about the occluding object can be used to postulate the position and orientation of the occluded object.

We can summarize the characteristics of the tactile domain with the following list of attributes. These can be used to make explicit assumptions about the domain and thereby reduce the complexity of the processing algorithms.

* explicit three-dimensional information
* no specularity (compared with vision)
* no shadows (compared with vision)
* there are no scaling distortions (compared with vision)
* there are no perspective distortions (compared with vision)
* ease of separation of objects from background
* the information provides a local "view" of the environment
* the act of sensing may produce a physical distortion of stimulus
* occlusion assumes a form different from that in vision

What one has, then, is a sensory modality which provides a very limited view of the environment both in terms of the amount of the environment sensed as well as the spatial resolution of the sampling. Yet this information has some very interesting characteristics. Unlike any form of visual sensing, including ranging, there are none of the problems associated with the processing of a two-dimensional image of an inherently three-dimensional world. Since the sensor is a contact device, the image received from the sensor is an exact indication of the state of the interface between the stimulus and the transducer. This also introduces problems in that the act of sensing may affect the stimulus, and the local nature of the information implies that the system using the tactile information must maintain a model of the spatial position and orientation of the tactile sensor relative to the other items of importance in the workspace.

The next logical step involves understanding the processing of tactile images. That is, how can the characteristics of the static and dynamic tactile data domains be used to construct simple algorithms for the extraction of information from the images? The next section discusses several algorithms and points concerning static tactile image processing.

4.4 Static Tactile Image Processing Experiments

The purpose of robotics tactile sensing is to provide information regarding the interaction between the robot fingertip and the world. This information may include the identity of a stimulus object or object feature, the position and orientation of the stimulus or feature of the stimulus, or the parameters of the interaction, e.g. the forces present. The review section above discussed numerous efforts in the area of tactile image processing. Just previous to this section, the characteristics of the tactile domain were outlined. This section discusses several

experiments which draw upon these characteristics to simplify the design of processing algorithms for tactile image data.

Two basic approaches can be taken toward tactile image processing. The class to which a particular algorithm belongs depends upon the information desired by the algorithm and the level of abstraction from the image data of the algorithm. One approach may be termed "data driven" and refers to the case where the ultimate information of interest is obtained directly from the image. A second approach is "extracted-feature driven" and refers to the situation wherein features such as edges and holes are extracted for the tactile image and are then used by another process for, say, object identification. If the goal of a particular algorithm is to detect the boundary of an object, it may use an edge finding procedure and would be called a data-driven algorithm. Another algorithm could use the same edge information as input but integrate the simple edge data into the closed contours of the object and from those contours extract the identity or position and orientation of the stimulus. This type of algorithm would be termed an extracted-feature algorithm.

The experiments presented below are not intended to provide an exhaustive treatment of tactile image processing but rather to demonstrate the usefulness of tactile array information and the simplicity in algorithm design that can be obtained by utilizing the characteristics of the domain. Data-driven as well as extracted-feature driven algorithms are demonstrated.

One of the first image processing techniques developed for visual image processing in an industrial domain used binary images produced by back lighting of the parts. These silhouettes were typically produced by viewing the object from above while it rested on a light table. The resulting binary images were processed by a system which collected the statistics for each of the regions or "blobs" in the image. These statistics were saved during a learning phase and matched against the statistics for regions seen during the actual running of the system. As a first attempt at tactile image processing one may be tempted to take this approach. Consideration of the characteristics of the domain, however, precludes this approach.

Recall that vision is global in nature in that it produces an image which is a two-dimensional version of the three-dimensional world. The tactile array sensor also produces a two-dimensional image of the world but since the sense is a contact sense, the image only includes what is in contact with the device. What this means in a practical sense is that in the case of vision, the entire object can be in the field of view, while in the tactile case usually only a very small section of the object is in view. In addition, the generic vision case consists of the object being viewed resting in a stable state depending upon the supporting surface and gravity. Since the camera does not affect the object, and most objects have a finite number of stable states, the number of possible views, excluding rotation about an axis perpendicular to the supporting plane, is limited. In the tactile case, however, not only does the act of sensing affect the object but the object is not constrained during grasping by a force such as gravity which is constant from instant to instant. It does not follow that one can expect the

same tactile "views" of an object grasped at two different times in the same way that one can expect similar visual images. Another consideration is the low spatial resolution of the tactile array sensor which dictates that fine details of an object's surface are not detectable, and that only those features which are large relative to the sensor surface are discernible.

For these reasons an experiment utilizing a blob approach to tactile image processing is not included here. The experiments outlined below work with the tactile images as grey scale images and, in some cases, extract low-level features such as edges and in other cases high-level, abstract features. The high-level features dealt with usually appear only once in an image. These features can then be used by a higher level process for object identification or pose determination.

4.4.1 E1: Edge Finding

As indicated earlier, one use for tactile array sensor data is for the location of the edges of a stimulus. The fact that the tactile sensor is a contact device implies that edges of an object which are in contact with the sensor bear a one-to-one correspondence with changes in the distribution of the forces on sensor surface. Therefore, finding the edges in a tactile image corresponds to finding the actual edges of the stimulus. The technique used here is from Hanson and Riseman [89] and proceeds in two steps.

The first step consists of extracting raw indicators of edges. This is accomplished by convolving the tactile image with a 1x2 difference operator oriented in all four directions. This results in an image containing the differences between adjacent forcels. These data are used as the initial estimates of the probability of the existence of a edge between the forcel pairs. For example, if the intensities of two given, adjacent forcels are very different, then the magnitude of the difference will be large, signaling the possibility of an edge. A probabilistic relaxation process which contains a number of rules used to update the probabilities is applied to these initial estimates. For instance, high probability edges aligned end to end and colinear support one another while a low probability edge parallel to and one forcel away from a low probability edge will cause the low probability edge to be suppressed. Figure 4.1 contains four raw tactile images from the 10x10 tactile array sensor prior to any processing.

The results of 10 iterations of the relaxation process on the edges produces the images shown in Figure 4.2. From these thinned edge images the boundaries of the objects can be extracted by finding the long chains of edges. This information can be used to try to identify the stimulus, and more simply, to orient the gripper with respect to the object more accurately. Any of a number of edge detection techniques can be employed, e.g. [36], to extract such information. The important point is that the tactile data include easily extractable edge information which corresponds directly to features of the stimulus object.

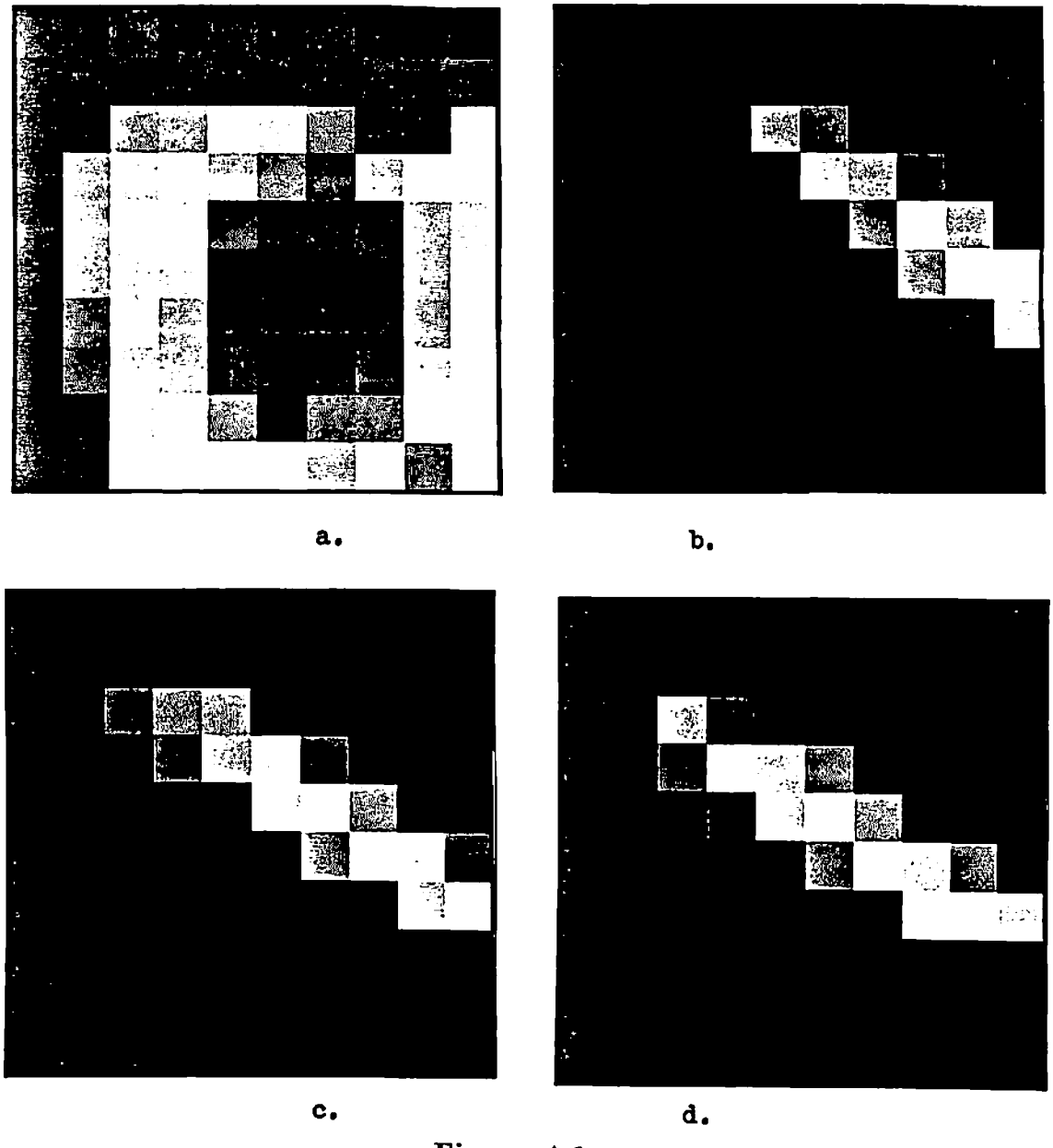


Figure 4.1

This figure contains four raw tactile images. The images were obtained from the Version 1 sensor and have had no preprocessing applied. 4.1a: image from a small plate with a hole. 4.1b,c,and d: three images from a sequence of a 0.125″ diameter cylinder moving across the sensor.

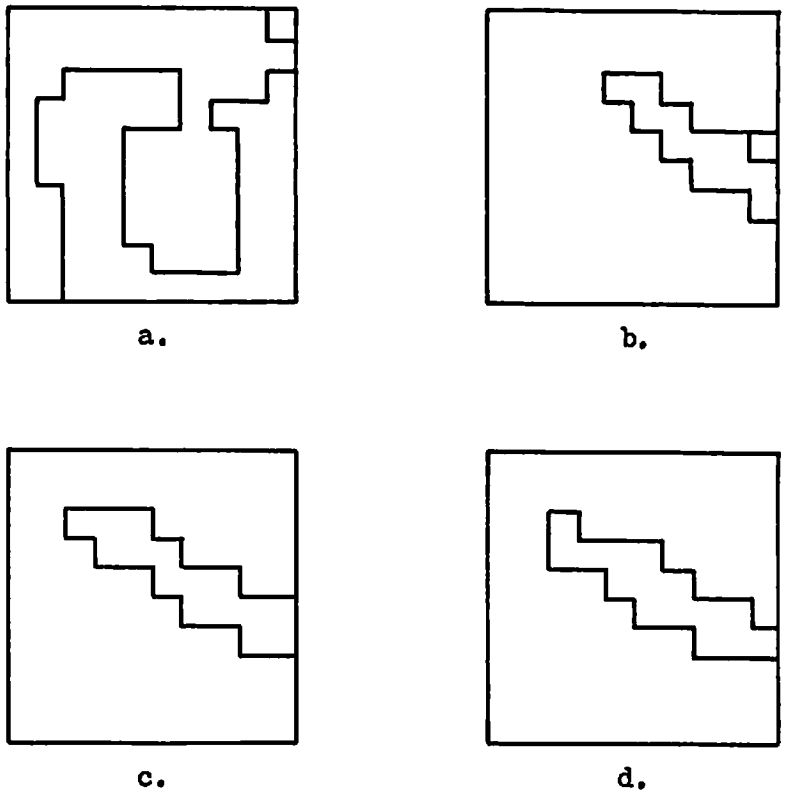


Figure 4.2

These four images are the edge images corresponding to the data images shown in Figure 4.1. A simple 1x2 difference operator was used to generate the initial edge images. These images were used as initial probabilities in a relaxation labeling process. The results after 10 iterations are shown here.

4.4.2 E2: Ridge Estimator

This experiment illustrates a technique for extracting a feature from a tactile array information. The feature of interest in this case is the axis of a cylindrical stimulus. This feature then contains information regarding the orientation and position parameters of the cylinder being scanned.

The characteristics of the tactile domain as discussed above can be used to simplify this algorithm. The contact nature of the sensor and the associated lack of scaling and perspective problems implies that the characteristics of the stimulus project directly into image characteristics. For example, it is assumed that the ratio of the major axis length to the diameter of the cylinder is large

enough that the image contains a ridge of high force values as opposed to a simple peak. The algorithm processes the image to extract this ridge. In addition, since the sensor provides a very localized view of the environment, it is reasonable to assume that only one stimulus object is in contact with the sensor and, therefore, all activity on the sensor is caused by the same stimulus. Once found, a line is fitted to the ridge and the set of points defining the ridge checked to see that it does correlate with a line. This algorithm is simple in nature and intended only to demonstrate the fact that the sensor data contain useful information.

The algorithm scans the image both horizontally along each row and vertically along each column. For each row and column scan, the maximum value is found. If the value is above a threshold defined as a percentage of the range of the image, then the row and column coordinates of the forcel are saved in a list. Once all rows and columns have been scanned, the list of saved forcel coordinates is thinned so that no coordinate pair appears twice. The remaining points are assumed to be representative of the ridge. Figure 4.3 shows three images produced by a cylindrical stimulus and the elements chosen by the algorithm. The chosen forcels are delimited by a surrounding box.

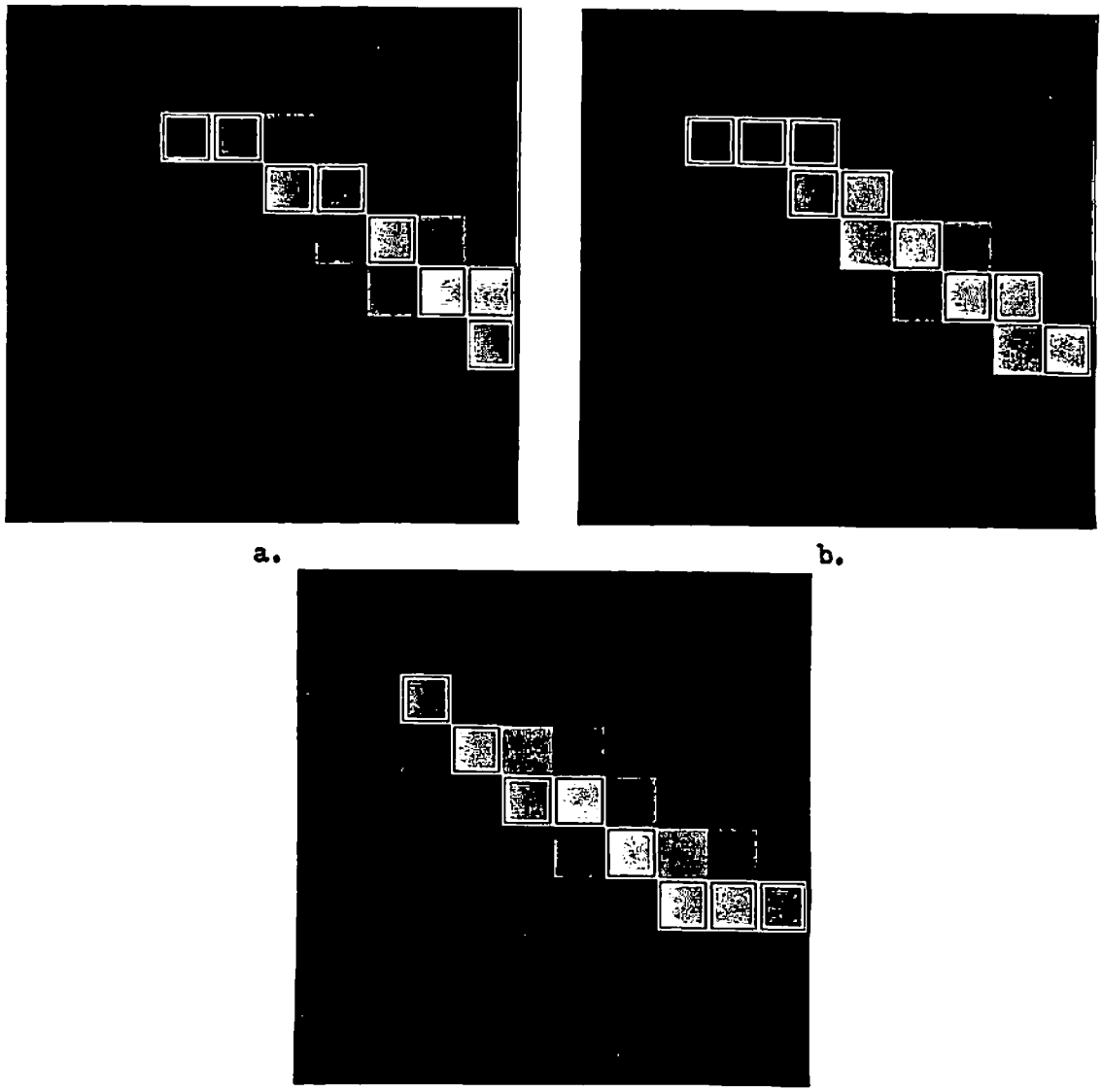
Linear regression is performed on the points to determine the best-fitting line. This line is used as the estimate of the orientation of the axis of the cylinder. The Pearson correlation coefficient, r, is calculated and no estimate is given if $r<0.5$, indicating that the line does not accurately fit the data.

The extent of the cylinder is determined by finding the points on the regression line which are the closest together such that all of the selected points in the image fall between the two lines perpendicular to the regression line passing through these points. Figure 4.4 shows the same set of images as in Figure 4.3 with the estimate of the axis overlaid on the image. Note that the width of the overlaid bar is for display purposes only and does not indicate an estimated width.

The length of the cylinder can be estimated as the distance between the chosen endpoints and the orientation is given by the angle of the regression line with respect to some axis. Table 4 contains the values of the parameters of the line as calculated by this procedure.

It should be noted that the cylinder was not completely on the sensor when the first image, cf. Figure 4.3a, was collected. The length estimate produced for this image is less than the estimates for the second and third images. Note that the angle estimates for all three images as well as the length estimates for the final two images are very consistent.

This type of information is quite useful to machines which manipulate objects in an environment which may not be completely known. For example, it may be necessary for a robot to use a long thin tool, e.g. a probe. After the tool has been grasped, tactile array information processed by an algorithm similar to the one outlined here could yield the position and orientation information needed to effectively use the tool. Further, this information can be monitored over time to extract motion and detect slippage, see Chapter 5. This type of

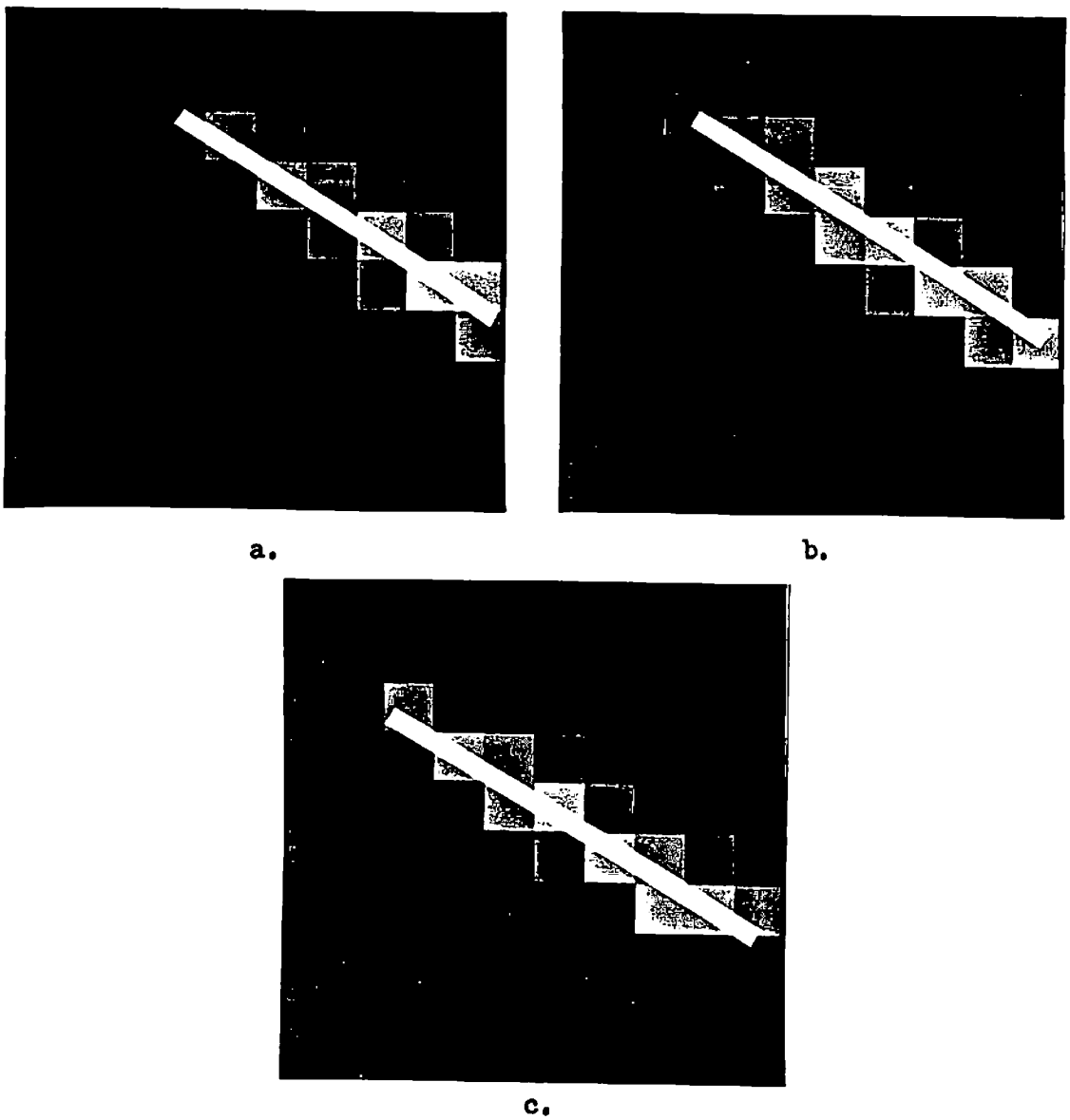


a. b.

c.

Figure 4.3

These images show the features selected by the ridge estimator process described in the text. The data images used were the cylinder images shown in figure 4.1b,c,and d with the bad forcels' values replaced by the average of the 3x3 neighborhood centered at the bad forcel. The bad forcels were determined automatically as those forcels which showed no variation in output over a set of randomly chosen images.



a. b.

c.

Figure 4.4

This figure contains the original data images with the estimate of the location and extent of the ridge overlaid. The width of the ridge estimate was chosen only for reasons of display clarity and has no relevance to the width of the stimulus.

Image #	Endpoint x	Endpoint y	Endpoint x	Endpoint y	Midpoint x	Midpoint y	Length (inches)	Angle (degrees)
1	0.98	0.39	0.35	0.79	0.67	0.59	7.5	32
2	0.97	0.36	0.26	0.79	0.62	0.57	8.25	32
3	0.94	0.30	0.21	0.73	0.58	0.51	8.5	31

Table 4

This table contains the values of the parameters calculated by the ridge estimator algorithm for the images in Figure 5.3. The parameters include the endpoint and midpoint locations, the length of the ridge, and the angle of the major axis.

experiment was performed with the sensor mounted on one fingertip of a two-fingered, parallel-jaw gripper.

The gripper control program began with the jaws in a completely open configuration and closed them until a prespecified force threshold was detected. The applied force was obtained by summing the outputs of the elements in the tactile array sensor. Once the threshold was reached, the closing motion was terminated and the ridge estimation algorithm begun. In this case, the selected points for the algorithm were generated by simply thresholding the force image. All forcels above the threshold were used in the regression calculation. Figure 4.5 shows the hardware configuration and the output of the system. On the screen is displayed the binary image resulting from the thresholding and the estimator.

Several points should be noted about this experiment. First, the feature used was extremely simple and the discrimination measure used equally simple. It was assumed that only one stimulus object was in contact with the sensor at the time that the image was collected. Multiple stimulus objects or an inappropriately chosen threshold value would result in either no estimate or a very poor estimate. It is not intended that such an algorithm be relied upon as the only source of information but rather that it should serve a role as one of a number of feature extraction algorithms which simultaneously process the images. The results of these algorithms are weighted and combined by a control process which ultimately decides which features are most likely to exist. The succeeding experiment illustrates another feature-extraction algorithm.

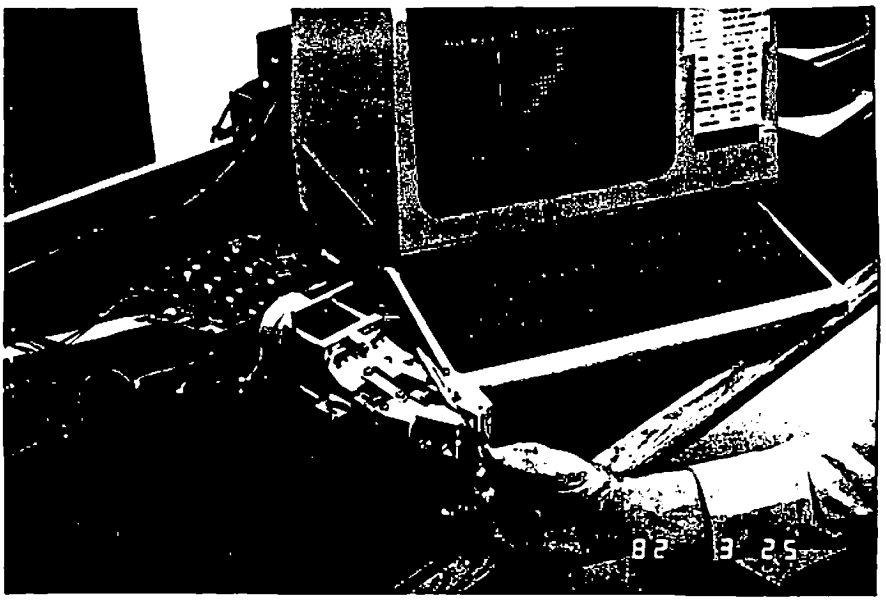


Figure 4.5

This shows the experimental set-up used for both this experiment and the object identification experiment. A tactile array sensor is attached to one fingertip of a two-finger, parallel-jaw gripper based on the URI design. A binary tactile image is being displayed on the screen. The gripper is holding a long thin object, a pen, and the angle of the object is clearly visible on the screen.

4.4.3 E3: Hole Center Finding

In many robot applications, it is necessary for the robot to insert one object into a hole in a second object. Exact position information regarding the hole may not be available. In this case either a mechanical solution such as a remote center compliance device or sensory information must be used. Mechanical solutions are applicable only in the case where the location is known to within the usable range of the device; thus, they are inappropriate in cases where no location information is available. Many sensory modalities can be used to locate a hole in a surface but this experiment deals exclusively with tactile array information. This algorithm was designed to find the center of a circular depression or peak in a tactile image.

Figure 4.6 contains a grey-scale representation of a force image obtained from the 10x10 version of the tactile sensor array. The algorithm described here attempts to find the circular region in the lower right section of the image.

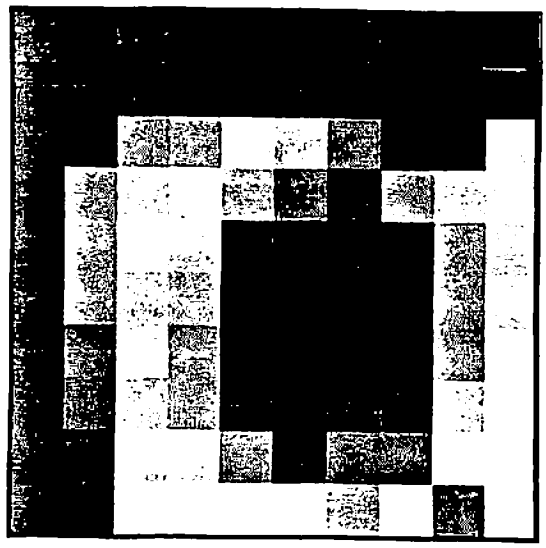


Figure 4.6

This image is a grey-scale representation of a force image obtained from the 10x10, Version 1, tactile array sensor. The stimulus was a small plate with a hole in the center. The interest is in developing an algorithm which can find the circular region in the lower right section of the image.

This algorithm relies upon two basic assumptions: i) a circular feature is located somewhere in the image and ii) the circle has a fixed, pre-specified radius. The assumption of a known radius for the circle is reasonable since it is typically the case that the parameters of the hole are known with the exception of its location relative to the manipulator. The characteristics of the tactile domain allow this knowledge to be translated directly into masks which can be applied to the tactile images. A coordinate pair is returned every time an image is processed, regardless of whether or not there is actually a circular region in the image. In addition, it searches for regions of a specific size; it may or may not find holes of different sizes. Discussed below are extensions to the algorithm which alleviate these assumptions.

Once collected, the tactile image is scaled so that all of its values fall in the range $[0,100]$. Multiple thresholding is used to break the image into sets of regions at various percentages of full scale. With the image scaled to a pre-specified range, the multiple thresholding is easier since a given threshold increment always produces the same number of binary images.

The scaled image is thresholded at several equally spaced levels to produce an initial set of binary images. Figure 4.7 contains the initial set of binary images produced for the image in Figure 4.7. The thresholds used range from

5% to 95% of full scale in increments of 5%. For a given threshold value, k, the binary image produced consists of only zeros and ones. A forcel value greater than or equal to k in the scaled image will result in a 1 being assigned to the corresponding forcel in the binary image. Similarly, a value less than k will result in 0 in the binary image. The set of binary images, one for each threshold value, is used by the remaining sections of the algorithm.

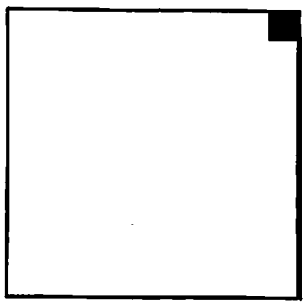
Each binary image is processed in turn to produce a center location estimate for that threshold value. The first step in this processing consists of uniquely labeling all 4-connected regions in the binary image. Figure 4.8 contains the labeled regions corresponding to the binary images in Figure 4.7 beginning with a threshold level of 40% of full scale. The 4-connected images for the thresholds below 40% are trivially inferred from the first image in Figure 4.8. Forcels are in the same 4-connected region if they have equal values and are adjacent either horizontally or vertically. Two forcels which are 4-connected to the same force are also 4-connected to one another. For each region, the coordinates of the centroid are calculated and a circle centered at the centroid is fitted to the region. Since the size of the hole is known, a mask corresponding to that size can be generated. This mask is overlaid onto each region such that its center is coincident with the location of the centroid of the region. A quantity describing the error of fit of the circle to the region is calculated based on the total area on the non-intersecting sections of the circle mask and the region being fitted. The center location and error of the best-fitting circle are saved and used to represent that threshold level.

Once all of the threshold levels have been processed, the level of the region exhibiting the smallest error is selected for the second pass. The second pass is identical to the first pass in that multiple thresholds are used; however, the threshold step is much smaller in the second pass and the range is centered at the threshold level chosen from the first pass. Furthermore it extends far enough above and below this value to include only the next adjacent threshold levels from the first pass. In the experiment at hand, the level chosen during the first pass was 65% of full scale. The second pass used threshold steps of 2% and covered the range for 59% to 71% of full scale. The labeled 4-connected regions from the second pass are shown in Figure 4.9.

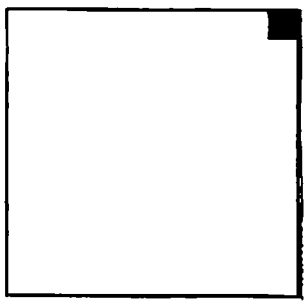
Each of the 4-connected regions at each threshold level is compared with the mask for the sought-after circle and the error measure calculated. The centroid of the best-fitting region is used as the estimate of the hole's location. Figure 4.10 shows the final estimate displayed in relation to the original image. The cross hairs indicate the location of the center of the circle and the purple ring shows the size of the circle used in the algorithm. The estimate appears to be relatively close to the actual center of the hole. The actual location of the hole in the stimulus object relative to the sensor was not recorded at the time the image was collected; thus, only subjective evaluation of the results of the algorithm is possible.

Figure 4.7

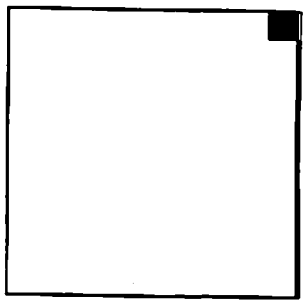
This is the initial set of images produced by the hole-finding algorithm. The raw data image in Figure 4.6 was scaled to have a range of [0,100]. These images are the binary images produced by thresholding the scaled image beginning at 10% of full scale in steps of 5%. The white areas are above threshold while the black areas are below.



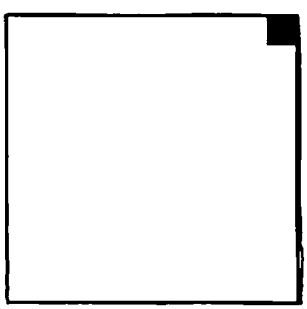
a.



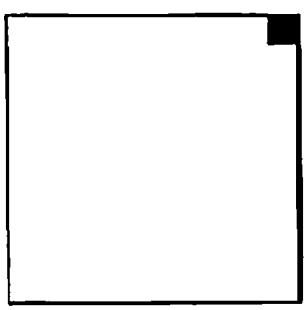
b.



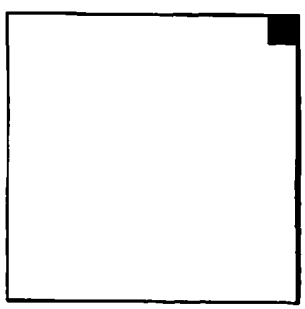
c.



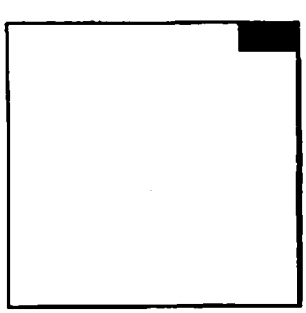
d.



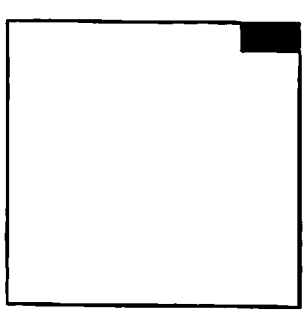
e.



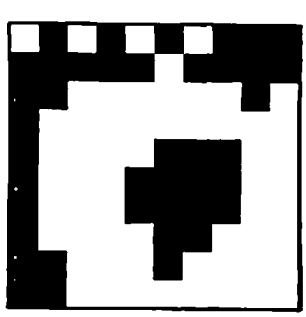
f.



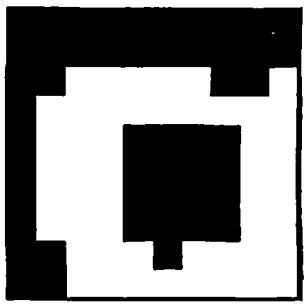
g.



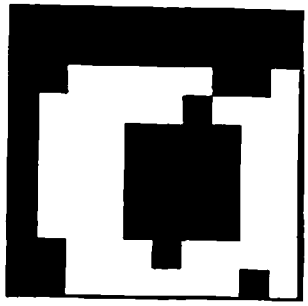
h.



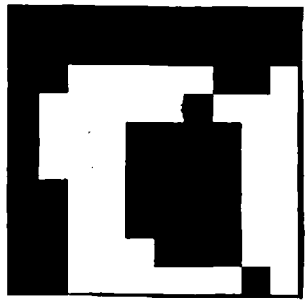
i.



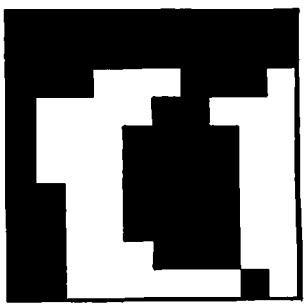
j.



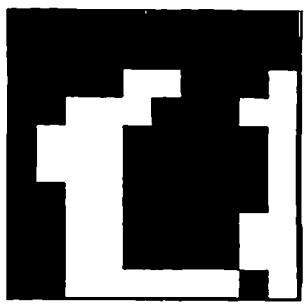
k.



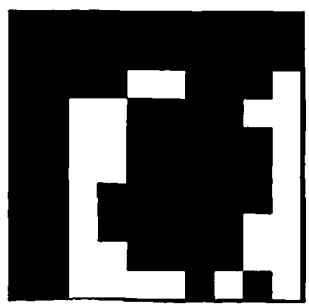
l.



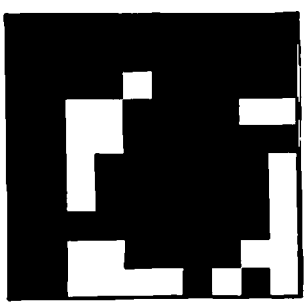
m.



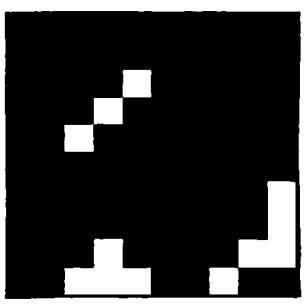
n.



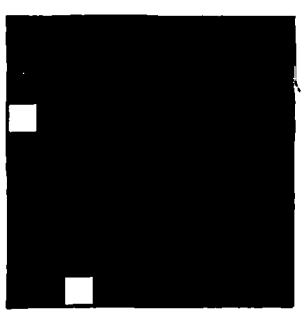
o.



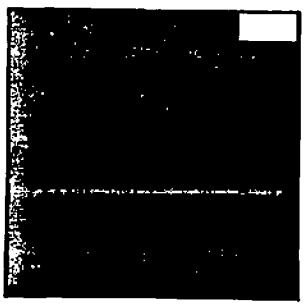
p.



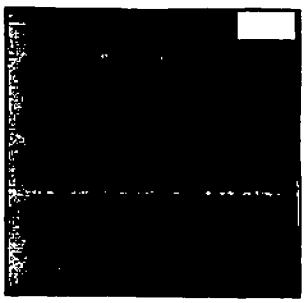
q.



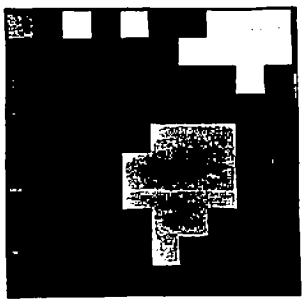
r.



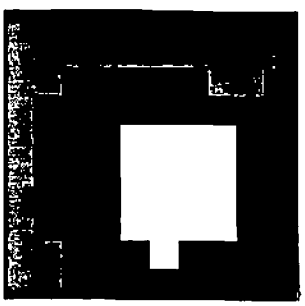
a..



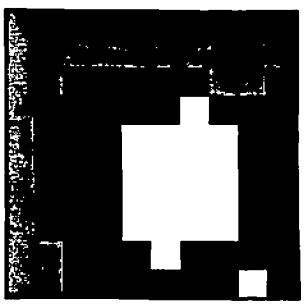
b.



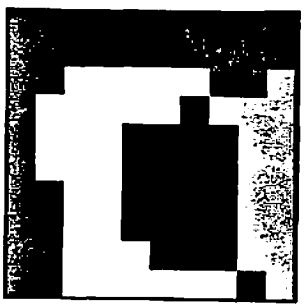
c.



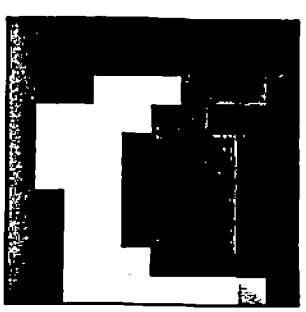
d.



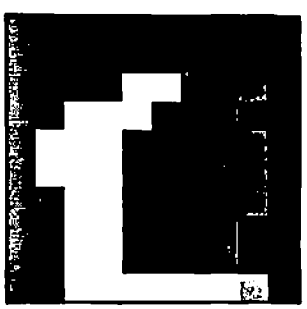
e.



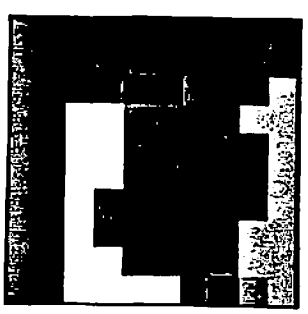
f.



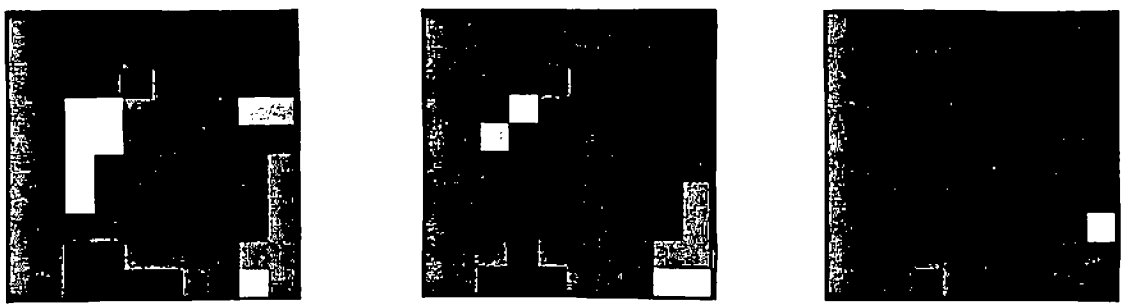
g.



h.



i.



j. k. l.

Figure 4.8

This is the set of 4-connected regions for the binary images in Figure 4.7 with thresholds equal to or greater than 40% of full scale. Forcels are in the same 4-connected region if they have equal values and are adjacent either horizontally or vertically. Two forcels which are 4-connected to the same force are also 4-connected to one another. A mask for a circular region is fitted to each region in these images and the centroid of the best-fitting region and its error measure are saved.

As mentioned in the introduction, this algorithm utilizes two assumptions. The first assumption, that there is a circular region somewhere in the image, results in an estimate being returned regardless of the degree of success in the fitting section. The output of the algorithm could be modified to include a quantity indicating the degree of fit of the circle. Such a quantity could be derived from the fit errors of the individual estimates at the various threshold levels. The second assumption, that the radius of the circle is known, allowed the algorithm to use a preset value for the radius for which it searched. The algorithm could easily be extended to search the radius space; and it could be applied at each threshold level for a range of radii, saving the center location, fit error, and radius of the best-fitting circle. The final stage would be modified to return the radius of the best-fitting circle as well as the centroid of the selected region.

It should be noted that the force images used for this algorithm, as with the previous one, use raw images with no preprocessing to remove non-linearities or noise. The point of this exercise was to demonstrate that raw data obtained from the sensor contains useful information.

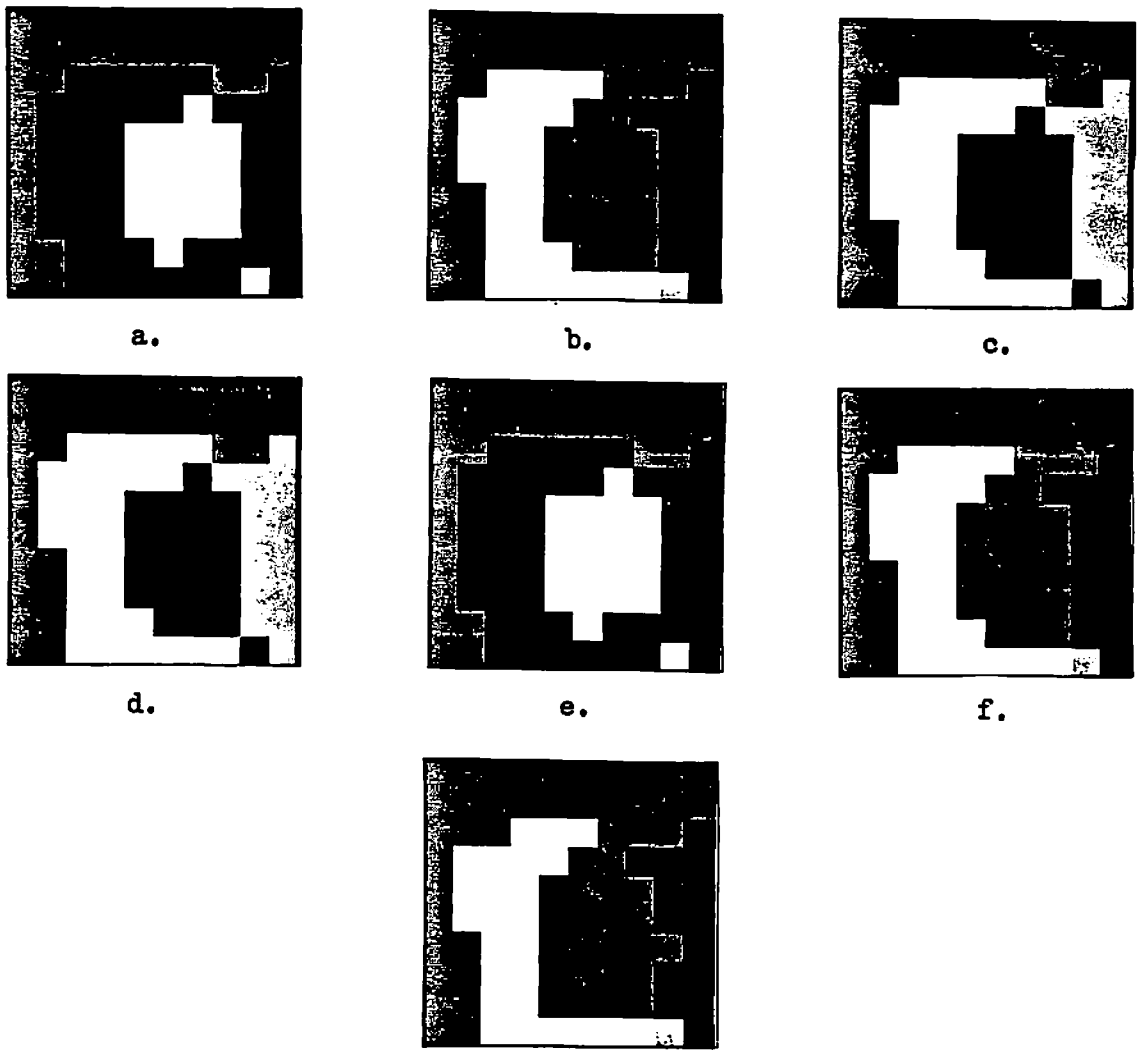


Figure 4.9

These regions were generated during the second pass of the hole-finding algorithm. The threshold level containing the region with the smallest error from the first pass was used as the starting point for the second pass. The threshold interval bracketing this value was subdivided and these new thresholds used to produce a second set of binary images. The 4-connected regions resulting from this new set of binary images is shown here. The thresholds range from 59% to 71% in steps of 2%. The centroid of the best-fitting regions in this set is used as the estimate for the location of the hole.

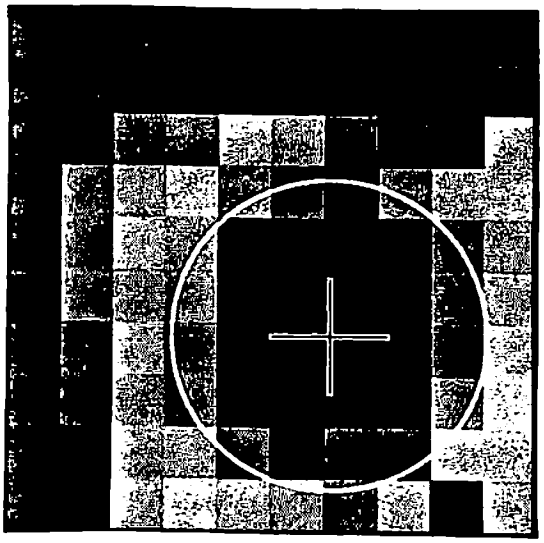


Figure 4.10

This figure shows the final estimate displayed in relation to the original image. The cross hairs indicate the location of the center of the circle and the ring shows the size of the circle used in the algorithm.

4.4.4 E4: Object Recognition

Experiment E2 used a data-driven approach to determine the orientation of the stimulus. Experiment E3 looked for a particular feature in the image. Information about this feature could be used by further processing algorithms to develop a model of the position and orientation of an object containing the hole(s). This algorithm is an attempt to perform object recognition from tactile sensation. The experimental setup was as described for the Experiment E2. A simple two-fingered, parallel-jaw gripper with a tactile array sensor on one fingertip provided the mechanical setup. A MINC system was used to control the gripper and collect the sensory information. This setup is the same as that shown in Figure 4.5. The information used by this algorithm is very similar to that used by Briot (see above) in that both a tactile image and separation of the fingertips are needed.

The purpose of this experiment was to demonstrate that the machine can recognize an object from touch. In particular, an unknown object can be grasped and, as the manipulator begins to move, a parallel process can interpret the tactile image in order to identify the manipulanda from the set of objects making up its experience.

The system has two phases: (1) a presentation phase in which each of the objects to be recognized in the future is presented to the gripper/tactile sensor

system and certain data stored, and (2) a run-time phase in which an object from the learned set is presented to the gripper and identified. During the presentation phase, the gripper closes until a force threshold is reached. The applied force is calculated by summing across the tactile sensor. When the threshold is reached, the closing action ceases and the sensory data are stored. The information stored includes a raw tactile image and the separation of the fingertips as derived from the position sensing system in the gripper. The user is queried for a name to be associated with the sensory information, the name and data stored, and the gripper opens to its starting position.

The run-time phase consists of the gripper closing on the unidentified object and loading a tactile image and the fingertip separation. The new sensory information is compared with the information stored during the presentation phase and picks the best match as the identity of the new object. The comparison consists of normalized cross correlation between the new and stored tactile images and differencing the new and stored fingertip separation. The correlation phase includes shifting the new image around in a small window to find the point of maximum correlation. The best correlation value and the separation difference (normalized) are combined in a weighted measure. The identity of the stored data producing the best match is used as the identity of the new object.

This algorithm was tested on a set of seven objects, three of which were transformers of approximately the same size but different coil/lamination orientations. Figure 4.11 shows the set of objects used. The transformers are of particular interest since they are functionally equivalent yet geometrically very different. In order to be manipulated properly, the identity must be known to provide access to the proper manipulation scheme. Figure 4.12 shows the system closed on one of the objects while it processes the sensory data. The system proved to be reasonably accurate in its identification of the objects. For some objects, discrimination was possible based on the fingertip separation alone but for others the tactile image data was required.

An obvious extension to this algorithm would allow for rotation of the object relative to the learned orientation. The current implementation can handle objects only when they are in roughly the same orientation as during the presentation phase. This technique was extended to more closely mirror the standard binary vision system approach; that is, the raw tactile image was thresholded and various statistics collected for the above-threshold region. The statistic vector calculated for each of the objects presented during the first phase was saved and used for comparison in the second stage.

All of the algorithms discussed have dealt with a single sensor and only one tactile image. Multiple sensors and multiple images provide some interesting additional constraints. Multiple images can be obtained in two ways. In the first case, the images can be collected at points separated temporally. This can be termed dynamic tactile sensing and is discussed in detail in the next chapter. The other case where multiple images can be collected occurs when more than one sensor is in use and the images are collected at spatially different but temporally identical points. This is shown schematically in Figure 4.13. In this

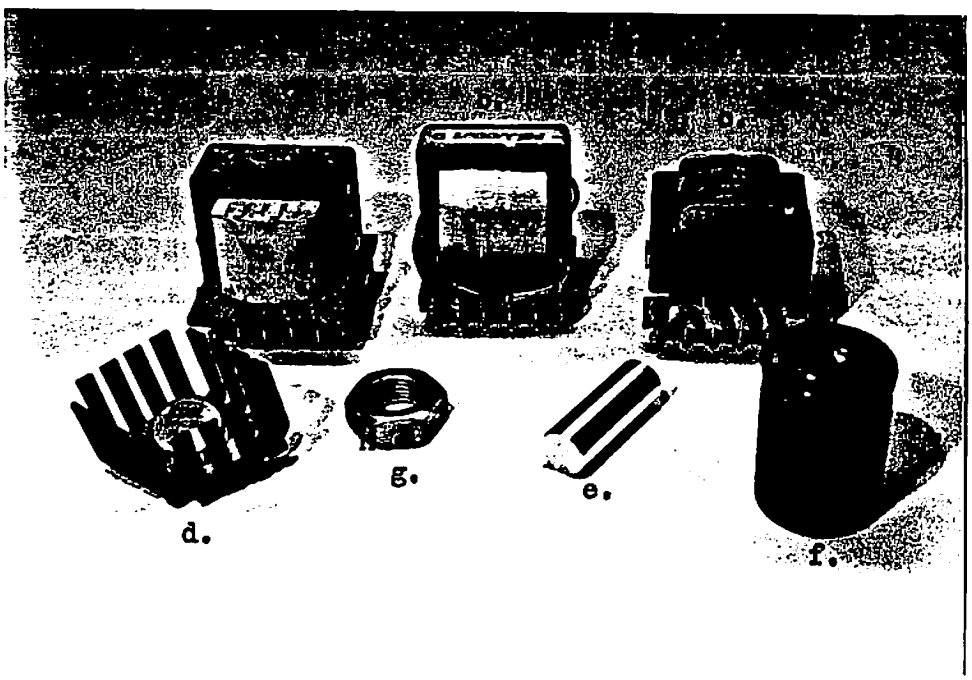


Figure 4.11

This contains photographs of the objects used in experiment E4. Objects a, b, and c are of interest because they are all transformers and have equivalent electrical properties. In addition, they appear very similar to a standard binary vision system. The variations in the orientations of the coils and laminations, however, cause them to produce different tactile images when grasped from a somewhat standard position. Object d is a power transistor with heat sink, e is an aluminum shaft, f is a capacitor, and g is a large nut.

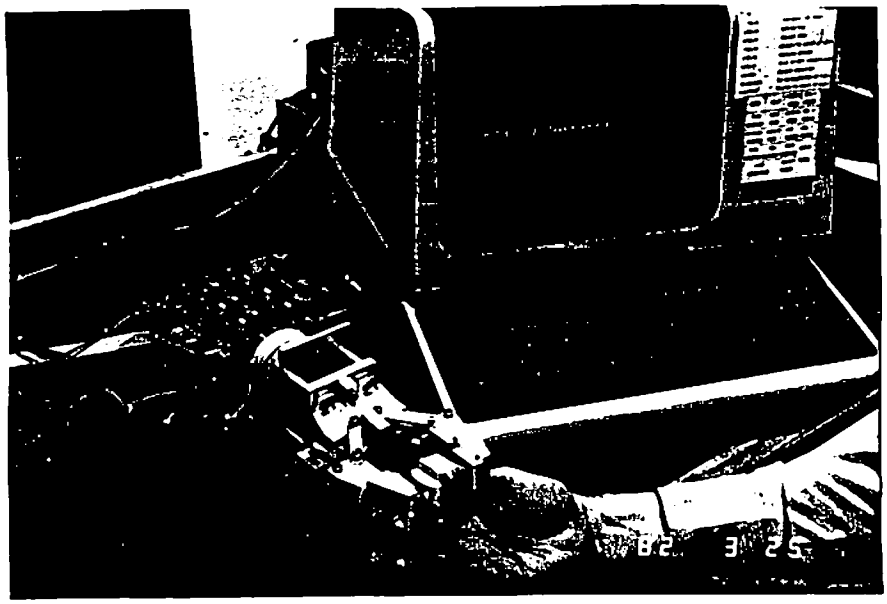


Figure 4.12

This shows the system closed on one of the transformers while the sensory data are processed. For some of the objects, discrimination was possible based on the fingertip separation alone but for others, e.g. the transformers, the tactile image data was required.

situation, the location of each sensor is known relative to the palm of the hand and thus relative to one another.

The result of this situation is that a number of very localized samplings of the object's surface are available from the multiple tactile images. In addition, the relationships between the local samplings is known. If it can be assumed that the stimulus is one object, then these relationships constrain the possible orientations of the local samplings relative to some object model. The same type of information can be obtained by sampling the tactile array information, moving the sensor, and resampling. From these data, a model of the surface of an object can be constructed. An example of this is given in the subsequent chapter.

This chapter has provided an overview of the current state of the tactile image processing. Several correction methods were presented for removing the

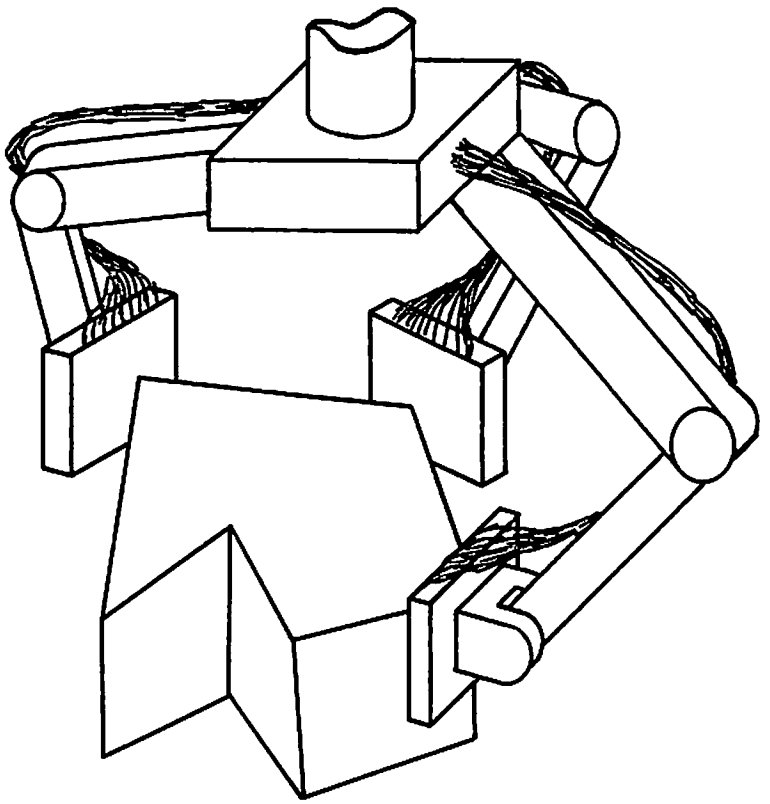


Figure 4.13

This shows a conceptual view of the use of multiple sensors. In this case, three sensors are employed, one at the tip of each of three fingers. The tactile images provide very localized views of the world while the kinesthetic information provided by the hand allows this local information to be integrated into one representation.

offset and response range differences inherent in the tactile sensor. The characteristics of the static tactile data domain were outlined and several experiments using algorithms based upon these characteristics presented. The intent here has been to provide a first step in understanding tactile image processing in the static case. The next chapter deals with processing in the dynamic case. The value of tactile sensing at its present state of evolution lies in the fact that gross features of stimulus objects can be extracted from the tactile images. The features can either be combined in an attempt to do object recognition or used by themselves to guide the manipulation of the object. Static

tactile sensing can be used for limited parts recognition as well as pose determination.

Chapter 5

Dynamic Tactile Image Processing

5.1 Introduction

Chapter 4 showed that information about the instantaneous location and orientation of objects and object features can be obtained from a static tactile image. Dynamic tactile sensation affords information concerning how these objects and features relate temporally to the sensor. For instance, the change in orientation or position of a grasped object as detected from changes in the tactile sensation is a clue that the object has slipped or otherwise changed its relationship to the sensor (gripper). In this case, dynamic tactile sensation refers to the act of passively monitoring the sensory input for changes, "passive touch". A more interesting approach occurs when the sensor is used to actively scan the environment. This form of sensing, often referred to as "active touch" [76], [98], and [16], employs the sensor to build an internal sensory data model of the environment. At any instant, previous sensory information is used to actively drive the device on which the sensor is mounted.

In this chapter, algorithms are described for passive touch rather than active touch since the data are processed after the entire sensory experience has been completed and the extracted information is not used in the control of the sensor. The algorithms are applied to a sequence of images derived from the tactile sensor (developed in Chapter 2) which provides a "snapshot" of the situation at the interface between the sensor and the stimulus object. Each tactile image is 16x16 grey-level force image, providing a static sample from a process which is continually changing. The dynamic nature of the interaction is captured by collecting a sequence of these static images which are separated temporally. The sequence is then processed to extract the desired information.

Dynamic tactile sensing paradigms can be divided into a number of classes. The first represents the temporal extension of the multiple-sensor situation outlined at the end of the previous chapter. In this case, tactile sensor data are collected over time from multiple sensors related by known kinematic transforms. Not only must the multiple tactile "views" from the individual sensors be combined, but these "compound" views from each sensor must be coalesced into a single representation. A second class involves restricting the information considered to that supplied by a single tactile sensor. Here the motion of the sensor relative to the stimulus and the features of the stimulus must be combined into a single representation. A sub-class of this involves the additional assumption that the multiple "views" provided by two consectutive tactile images have been produced by stimuli with a common structure. This assumption implies that the tactile images overlap on the stimulus and thus motion and surface model construction can proceed from information calculated relative to the images alone: without kinesthetic information relating the sensor position to one another via some world coordinate system.

The experiments discussed below employ the previously mentioned

assumption, and several of the experiments utilize data from the same experimental paradigm. In this case, the sensor was firmly attached to the bed of a Bridgeport vertical milling machine which allowed precise placement of the sensor in the x-y plane. The stimulus was placed in the head of the machine and lowered onto the sensor using the mechanism normally used to move the tool vertically. This arrangement allowed for rotational as well as vertical positioning of the stimulus.

The stimulus used was a cylindrical steel shaft 1/8 inch in diameter and 3/4 inch in length. Figure 5.1 contains schematic representations of the cylinder and sensor outlines and shows the actual position of the cylinder when each of the eight images in the sequence was collected. The position given below each image is the location of the center of the major axis of the cylinder in terms of the sensor coordinate system. Figure 5.2 contains the sequence of tactile images corresponding to the schematic representation in Figure 5.1.

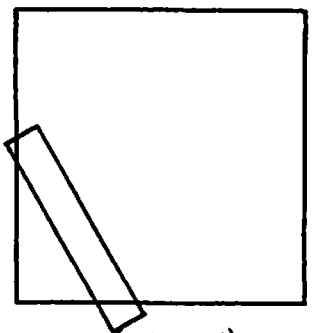
A basic problem when extracting information from a sequence of snapshots is one of matching. That is, what sensory event in image n+1 corresponds to a given event in image n? The events can be as rudimentary as individual forcels or as complex as highly processed semantic features. In the visual domain this matching problem has been referred to as the "correspondence problem" ([131], [110], [203] and in a more neurophysiological context as the "stimulus-matching problem" [37].) The images need not necessarily be separated in time for this problem to exist. For instance, temporally separated visual images are used primarily for motion and depth determination while spatial separation of visual images typifies stereo vision and poses similar matching problems.

In the general case, a particular image contains a number of points or features which are sought in the next frame. The motion of the object(s) being sensed is defined to be equivalent to the motion of the features representing the object. For a particular feature, several situations are possible: the feature may not appear in the next image due to occlusion or movement out of view, it may match with exactly one feature, or it may match with a number of features. Only in the second case is it possible to confidently assign a particular motion to the feature. When many features in one image are very similar, as is the case with the pixels as features, there may be a multitude of pairings possible and the difficulty arises when choosing among the alternatives. Typically a set of constraints is placed on the extracted motion in order to make the correspondence problem tractable, e.g. the motion over time must be smooth, cf. [168]. This constraint can be used to reduce the processing required by limiting the search space as is addressed below.

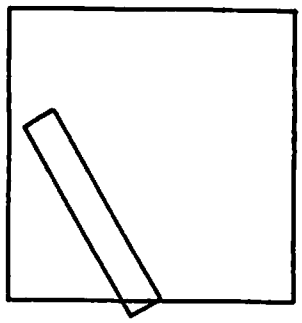
In some cases, the motion estimate for each image pair is calculated independently of any other image pair as illustrated in several of the experiments below. More sophisticated techniques employ the motion information extracted from earlier image pairs in the processing of later pairs [168]. A problem with such techniques exists in the case of the first few image pairs. This problem, known as the startup problem, involves the lack of information during the in-

Figure 5.1

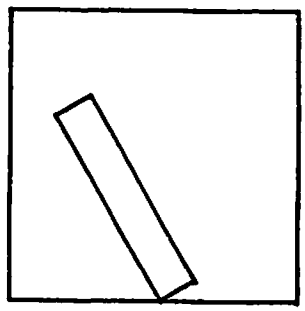
This contains a schematic representation of the position and orientation of the cylinder stimulus and the sensor. The outlines show the actual positions of the cylinder for each of the eight images in the sequence. The stimulus used was a cylindrical steel shaft 1/8 inch in diameter and 3/4 inch in length. The position given below each image consists of the location of the center of the major axis of the cylinder in terms of the sensor coordinate system.



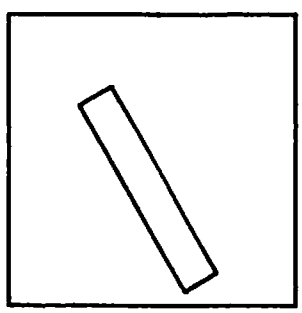
(.2,.25)
angle = 30
l = .750
w = .125
a.



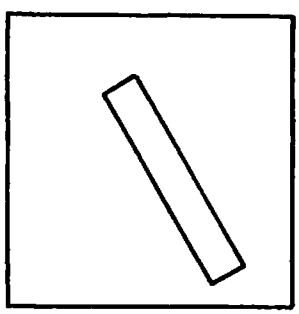
(.3,.3)
b.



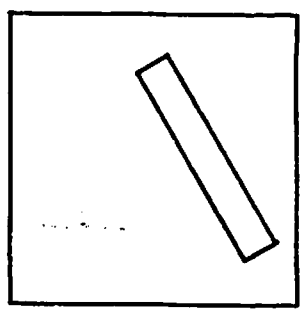
(.4,.35)
c.



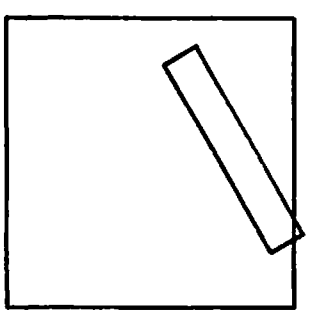
(.5,.4)
d.



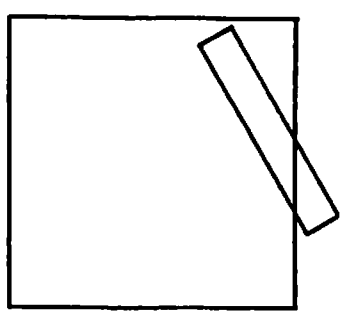
(.6,.45)
e.



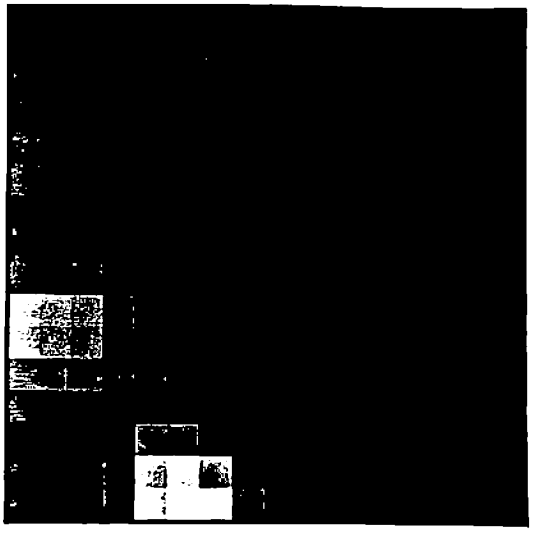
(.7,.5)
f.



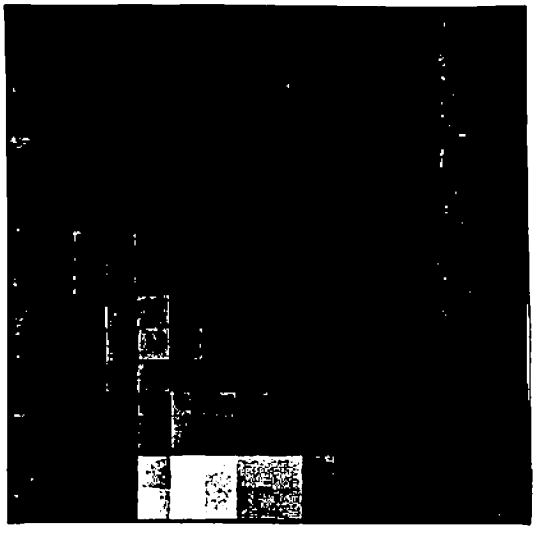
(.8,.55)
g.



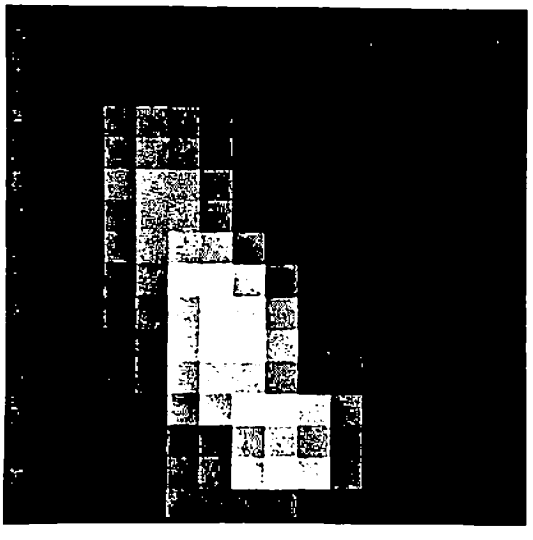
(.9,.6)
h.



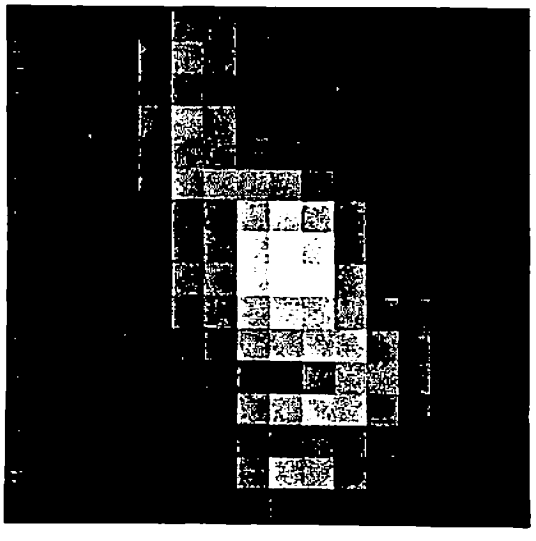
a.



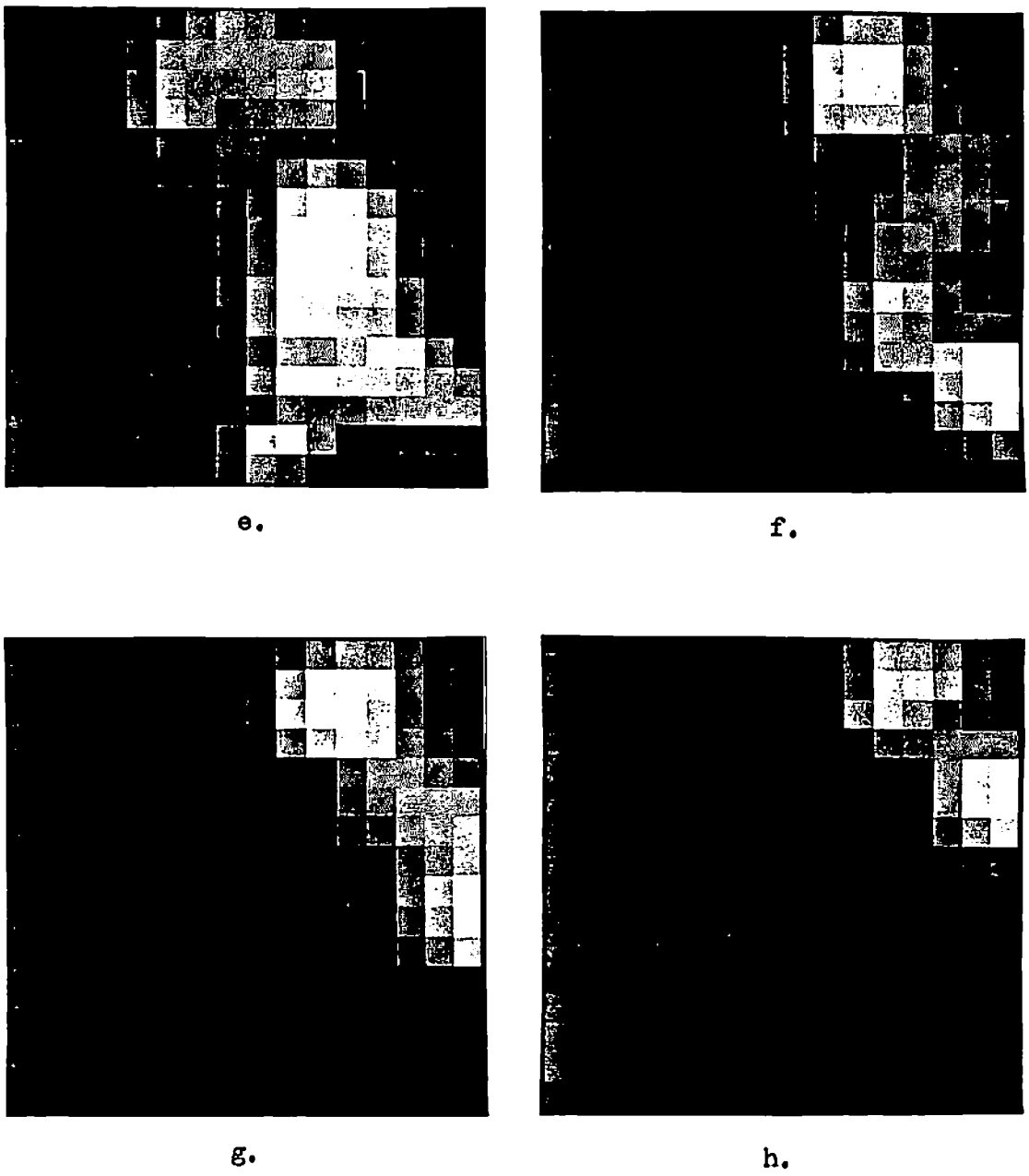
b.



c.



d.



e. f.

g. h.

Figure 5.2

These eight tactile images were collected under the experimental conditions diagrammed in Figure 5.1. The Version 3, 16x8 tactile array sensor was used and the images preprocessed with the "correction" method chosen in Chapter 3.

itial phases of the motion calculation. Without an indication of the motion, systems both require more computation to arrive at an estimate and can produce larger errors in their estimates than when the previous motion is known.

Dynamic machine vision algorithms have generally fallen into two categories. In the first, the objects in the image are identified in each frame and their image parameters and knowledge of their physical characteristics used to calculate the motion. In the second, the vector field created on the imaging surface by mapping moving features in the environment is processed to extract motion in terms of the parameters of the imaging system. This technique, known as "optic flow", [77] [78] [122] [124] has the advantage that motion information can be extracted prior to image interpretation. Image features are matched between image frames. The matching is guided by the fact that the general form of the motion of points is known, e.g. in the translational case points emanate from a single focus of expansion (FOE), travel along great circles when viewed on a spherical retina, and vanish at a single focus of contraction (FOC) [121], [79], [168].

The first approach is clearly applicable in the tactile domain. That is, objects can be identified in consecutive frames, and their motion can be determined. The optic flow approach does not, however, have a direct "tactile flow" analog. The movement of physical texture points across a tactile surface is similar to optic flow in that it produces a dense field of moving features. However, the fact that the tactile sense is a contact sense means that there are no points corresponding to the FOE and FOC of optic flow. Optic flow processes relax to a smooth velocity field which may contain the velocities of a number of objects. In the tactile case, a single velocity is of interest. This is due to the fact that typically only one object will contact the sensor at a time.

Two basic computational approaches have been taken for extracting information from image pairs: correlation and feature matching. Section 5.3.1 contains the results of several experiments based on cross-correlation; section 5.3.2 discusses a number of approaches to the extraction of motion information through feature matching between tactile image pairs. Notice that correlation is essentially an ad hoc feature matching process where the subimage in each "window" is treated as a feature.

Dynamic tactile sensing can be used to provide several useful forms of information. The most obvious of these forms is the motion of the object relative to the sensor. The motion of the sensor is typically known since it is immobile or fixed on a robot manipulator controlled by the system requesting the information. A useful application of dynamic tactile sensing is in slip detection. A part can slip in a robot gripper either because the gripping force is not sufficient to support the object's weight, or because the object being manipulated has hit an obstruction.

Another important use of dynamic tactile sensing is to construct a model of the object being sensed. In many cases, the entire object is not sensed at one time. Here, moving the sensor and collecting an image adds information about an area of the object's surface unavailable during the previous sampling. The

area of overlap can be used to register the images and thus a model of the surface can be built. Section 5.3.3 contains an experiment illustrating this use of tactile sensor data.

This chapter begins by reviewing the characteristics of the tactile domain defined in Chapter 4 and the additional characteristics and assumptions applicable when considering the dynamic nature of this domain. As indicated above, several experiments are presented which demonstrate the utility of the tactile sensory data both for determining the motion of a stimulus and for constructing a model of the stimulus surface. The final section provides a summary of the chapter geared toward possible directions for future work.

5.2 Domain Constraints

Before proceeding with a discussion of the experimental work, it is appropriate to summarize the characteristics of the tactile domain. We can recall from Chapter 4 the characteristics of the static tactile domain which also apply to the dynamic case. As before, these can be used to reduce the complexity of the processing algorithms.

* explicit three-dimensional information
* no specularity (compared with vision)
* no shadows (compared with vision)
* there are no scaling distortions (compared with vision)
* there are no perspective distortions (compared with vision)
* ease of separation of objects from background
* the information provides a local "view" of the environment
* occlusion assumes a form different from that in vision

On the negative side, it should be noted that:

* the act of sensing may produce a physical distortion of stimulus

In addition to these characteristics, we can add those specific to the dynamic domain.

* sensor motion known
* motion in sensor images directly corresponds to relative motion of the sensor and object

In the tactile domain, the motion on the sensor is exactly the motion of the object. Thus determining the motion of parts or features in the tactile images is equivalent to determining the motion of the object. Note, however, that an object could roll across the sensor, e.g. a cylinder, and the motion extracted from the data would indicate only the translational component and not the rotational component. This is similar to the case in vision of an object rotating about its axis of symmetry. If no texture information is available, the rotational

aspect of the object's motion is lost. The problem is fundamentally the same in the tactile domain. Without surface information, sliding of an object versus rolling of an object cannot be detected by processing images of the distribution of normal forces or deflections. If shear information is made available, then differentiation between the two situations may be possible.

5.3 Experiments in Dynamic Tactile Image Processing

In robotics, a special case arises when the characteristic of interest is the actual motion relative to the sensor and, as in holding a screw driver, no relative motion is desired. The changes from one image to the next should be very small and within the noise levels inherent in the device. This case corresponds to slip detection. Unfortunately, the object must have moved in order for the slip to be detected in this manner. That is, incipient slip is undetectable from strictly normal force or deflection information.

As with the static tactile image processing discussed in the previous chapter, two basic approaches to the processing of image sequences can be taken. The first of these may be termed "data-driven." In this case, successive images are compared directly. The second approach relies on the extraction of features from the images and the matching of these features from one image to the next. This approach may be termed the "extracted-feature-driven" approach.

5.3.1 Motion: Cross-Correlation Techniques

The first three experiments use cross-correlation techniques. E1 uses normalized cross-correlation to estimate the motion of a stimulus relative to the sensor. E2 extends the basic technique by restricting the area considered during the correlation process. E3 utilizes interpolated data to allow sub-forcel offset vectors to be chosen by the process.

5.3.1.1 E1: Basic Cross-Correlation

This experiment utilizes a data-driven approach, in that image n in the sequence is overlaid onto image n+1, and the value for the normalized cross-correlation between the two images is calculated for different relative positions of the two images. The result is an image of cross-correlation values for each pair of images. Each of these cross-correlation images consists of an array of normalized cross-correlation values with the location in the array corresponding to the offset from one image to the next. That is, image n was offset in every possible position such that there remained a non-empty overlap with image n+1, then the value of the normalized cross-correlation for the area of overlap were computed and stored at the location in the cross-correlation image corresponding to that (dx,dy) offset vector.

This use of cross-correlation rests on two assumptions in addition to those inherent in the domain. First, this technique searches only the space of translational changes between the images; rotational changes are not taken into account. Rotation adds one parameter to the search space and thus complicates the process. Secondly, it is assumed that only one rigid object is being felt at

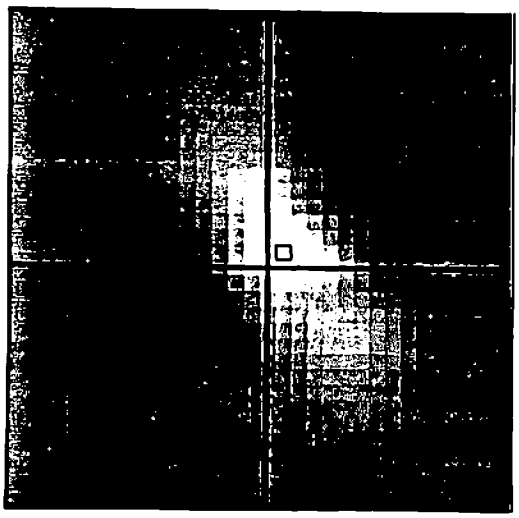
any given instant. This implies that all points on an object move with the same velocity vector for translational motion and about a common point during rotational motion. This assumption is usually correct due to the size of the sensor relative to the size of a typical stimulus object.

Figure 5.3 contains the results of this experiment. The seven images represent the normalized cross-correlation images between the pairs of consecutive images, i.e. between images 1 and 2, 2 and 3, etc. The axis system indicates the origin of the offset vector coordinate system. The range of correlation values is encoded using brightness with full positive correlation (+1) indicated by white, no correlation (0) indicated by grey, and full negative correlation (-1) by black. The element at the intersection of the axes corresponds to a zero offset vector. The element surrounded by the box corresponds to the offset vector which produced the maximum correlation value. This offset vector is taken as the calculated motion of the cylinder between the two images.

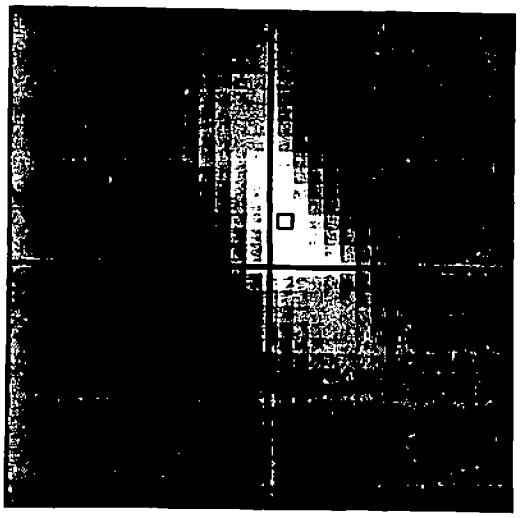
The appearance of the cross-correlation images is as one would expect from a correlation technique. Examination of Figure 5.3c, for example, reveals several interesting points. There is a ridge of positive correlation along the axis of the ridge seen in the original image. One can imagine the first image being slid over the second image and this ridge corresponds with the positions where the objects are in best correspondence. As the images are slid along the axis, the amount of overlap decreases and thus the correlation value decreases. When the images are slid along a line perpendicular to the axis of the positive correlation ridge, a region is encountered where the image of the cylinder in one image is matched with a background region in the other image. This results in a negative correlation value due to the fact that the correlation calculation is normalized using the mean value of the image intensities. Peripheral areas of the correlation image correspond to offset vectors resulting in overlap of only background areas.

An inherent feature of cross-correlation is that full correlation is possible only when the areas being used in the first image appear in the second image. This is not a problem for the middle images in the sequence since the object was completely covered by the sensor. At the beginning of the sequence, however, this is not the case. As the sequence progresses from the first image, the amount of the stimulus covered by the sensor increases; thus, there is a large area in the second image where the first image will fit and high correlation is possible over a large area of the overlap. Correspondingly, at the end of the sequence the amount of the stimulus covered by the sensor decreases with each image. As a result, the position yielding the maximum correlation value may not correspond to the proper spatial relationship between the images. This is evident in the sequence of motion estimates.

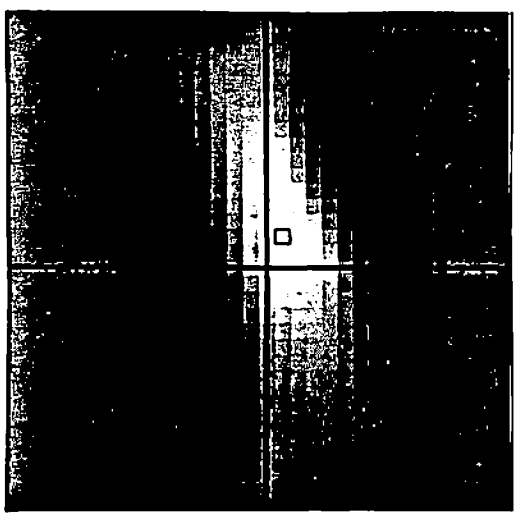
Sub-forcel interpolation for the cross-correlation was not used in this case (see below) and thus the individual x and y components of the offset vector are restricted to the integers. That is, the motion is restricted to integer numbers of forcels even though the actual motion was sub-forcel in magnitude. This results in rather choppy motion, but the trend is correct. Figure 5.4a contains



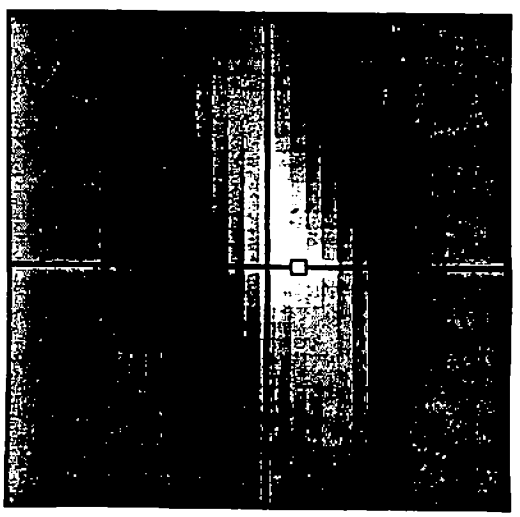
a.



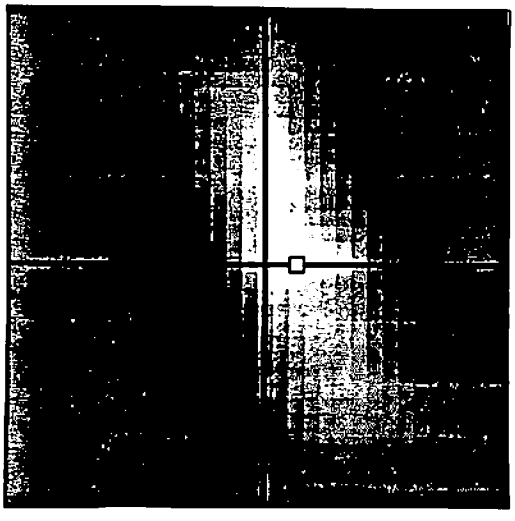
b.



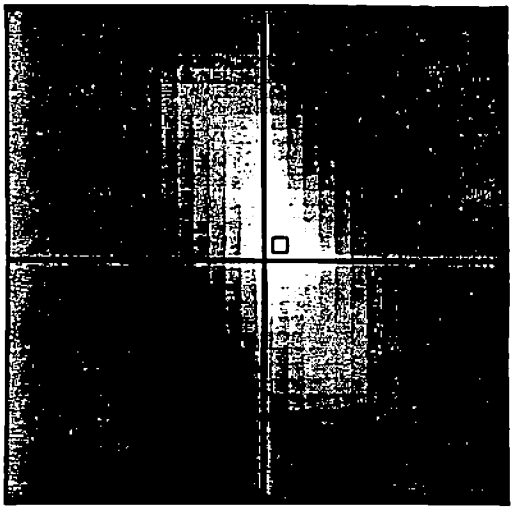
c.



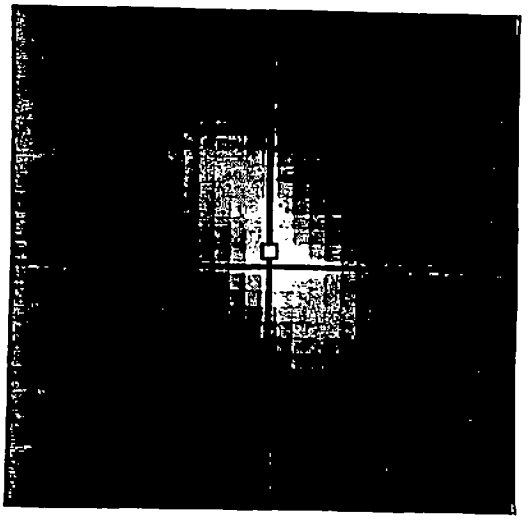
d.



e.



f.



g.

Figure 5.3

This figure contains the set of normalized cross-correlation images produced by overlaying the images with offsets corresponding to the locations in the correlation image. The seven images represent the normalized cross-correlation images between the pairs of consecutive images, i.e. between images 1 and 2, 2 and 3, etc. The axis system indicates the origin of the offset vector coordinate system. The element at the intersection of the axes corresponds to a zero offset vector. The range of correlation values is encoded using brightness with full positive correlation (+1) indicated by white, no correlation (0) indicated by grey, and full negative correlation (-1) by black. The element surrounded by the box corresponds to the offset vector which produced the maximum correlation value.

a schematic representation of the actual path of the cylinder while 5.4b,c and d contain the path as extracted by the normalized cross-correlation process.

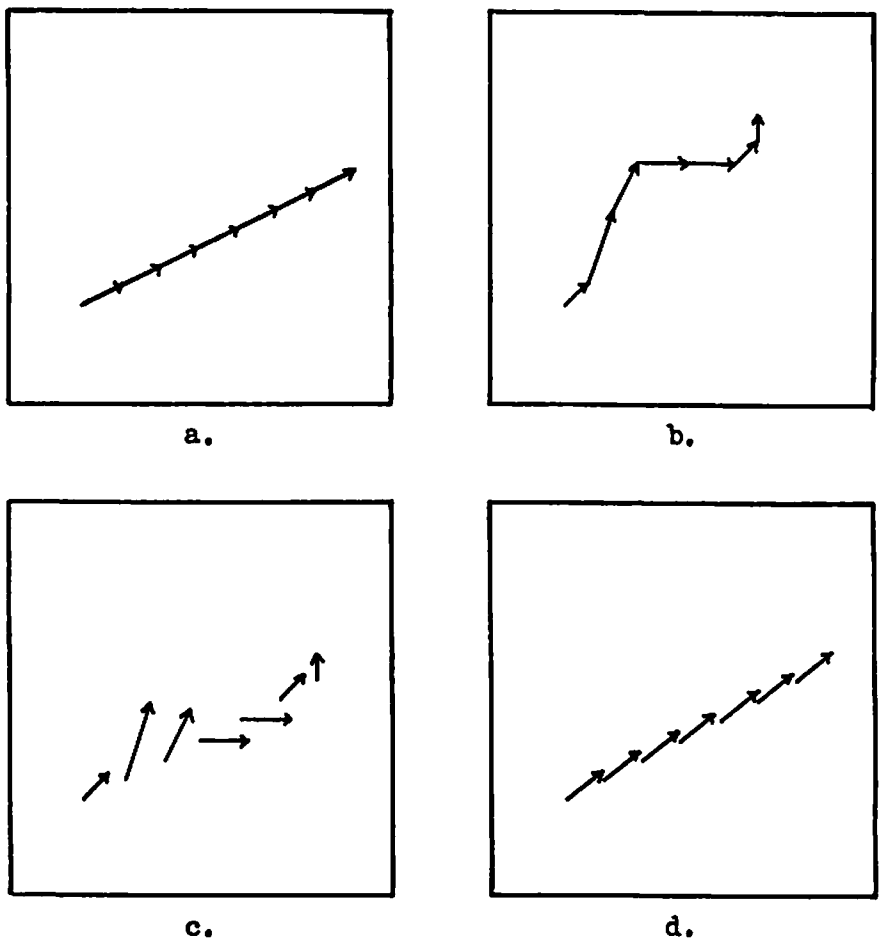


Figure 5.4

Figure 5.4a contains the actual cylinder path as shown graphically in Figure 5.1. The outline represents the border of the sensor and each symbol the middle of the cylinder axis. 5.4b, c, and d contain the path as extracted from the image sequence in Figure 5.2 by the cross-correlation process. The trace in 5.4b was produced by laying the calculated offset vectors head to tail beginning at the actual location of the cylinder in the first image. 5.4c shows the offsets placed at the actual locations, and 5.4d shows the average offsets placed at the actual locations.

The actual path of the cylinder is shown in Figure 5.4a, as given by the movement of the midpoint of the major axis of the object. The midpoint at each instant is indicated by the cross and the movement by the vector whose tail is located at the cross. The actual movement vector is (0.1,0.05) inches which corresponds to (1.6,0.8) forcels per image pair. The motion calculated from E1 is shown in Figure 5.4b, c, and d. In Figure 5.4b, the trace shows the set of offset vectors placed head to tail beginning at the actual location of the first cylinder. This method causes the errors in the calculation to accumulate as the sequence continues. The trace in 5.4c has the motion vectors displayed with each vector emanating from the actual location of the cylinder in that image. The quantized nature of the calculated offset vector is evident in this display. The final trace, 5.4d, contains the path defined by seven instances of the average offset vector.

Comparison of the predicted path to the actual path elucidates the errors made in the prediction. As discussed above, correlation techniques have inaccuracies when the two images in question do not represent the same extent of the object. This is emphasized by the errors at the end of the sequence. Other errors are introduced by the quantization of the offset vectors. In addition, the estimate produced for each image pair is calculated independently of the previous estimates.

Problems occur at the edges of the tactile images. Forcels along the edge of the sensor tend to have much more spontaneous variation than those in the interior of the pad. Since the output of some of these elements correlates poorly with the actual stimulus, the correlation of those forcels with the proper forcels in the next image is low. In addition, the stimulus exciting a given edge forcel may move off the sensor thus leaving the forcel with no corresponding forcel in the next image. The forcel will then correlate best with whichever forcel in the second image most closely matches. One method for dealing with these effects is to weigh the inner forcels more than the edge forcels in the ultimate motion calculation equation.

A common practice in machine vision motion systems involves using the previous motion estimates to either limit the search space in the processing of the current image pair or bias the estimates. Prager and Arbib [168] used the estimate of the nth optic flow field to initialize the processing of the n+1st image pair. A marked reduction in the time to convergence to the correct flow field was observed. A similar approach was investigated here which involved using the previous motion estimate to bias the estimation procedure.

The normalized cross-correlation was calculated as before. In addition, the previous estimate of the motion was provided to the system and the correlation value for each offset vector weighted by its Euclidean distance from the previous estimate. The distance weight employed was a linear function with a range of [0,1]. The function used a width equal to four times the width of the sensor, i.e. the distance weight for an offset vector equal to that predicted from the previous image pair is one and that for a vector greater than twice the sensor width away from the predicted vector is zero. The longest offset vector considered

by this process is a diagonal across the sensor, i.e. 1.4 times the sensor width. Thus the effective range of the distance weighting function is [.3,1] for the offset vectors calculated by this process.

The motivation for this weighting function involved the assumption of uniform translation for the motion of the object; therefore, the motion in the n+1st image pair should be the same as the motion in the nth image pair. Thus it is reasonable to use the estimated motion from one image pair as a prediction for the motion in the next image pair. (The approximation is still useful in a wider class of motions, so long as they are reasonably smooth.) The contact nature of the sensor implies that the motion felt is the actual motion of the object relative to the sensor. Since the motion is assumed to be smooth, it is likely that the offset vector calculated for a given image pair will be equal to the vector from the previous pair.

The estimate provided for a given image pair was calculated as the average of the estimates of the previous three image pairs when they exist. Figure 5.5 contains the results of the cross-correlation motion estimation process using the difference weighting with previous estimates. The path resulting from the individual estimates is shown as path a, cf. Figure 5.4a which contains the actual path. Averaging three consecutive estimates produces path b. Both of the paths are improvements over the original path estimates. Notice that the first estimate is the same as in Figure 5.4b, a result of the startup problem associated with such techniques. There is no estimate of the motion for the processing of the first image pair. The estimate "blindly" produced from the first pair is used as the prediction for the second pair.

The actual motion of the object was along a straight path inclined 0.46 radians from the x-axis with a speed of 1.79 forcels per image. This technique results in a calculated motion along an average path inclined .67 radians from the x-axis with an average speed of 1.83 forcels per image while the non-predictive method (Figure 5.4b) yields the less accurate result of .79 radians inclination and a speed of 1.62 forcels per image.

5.3.1.2 E2: Mask-Limited Cross-Correlation

Experiment E2 is the first extension to the previous experiment and uses a mask to limit the area of each image considered during the cross-correlation calculation. The aim is to restrict the calculation to only those areas of the images which are estimated to represent objects, thus eliminating influences from surrounding "background" areas. The mask for each image was produced by thresholding the image at a value corresponding to 25% of the full scale value for that image. The images produced for the masks are shown in Figure 5.6a-h where each element above the threshold appears in white. These binary images are used to direct the correlation process.

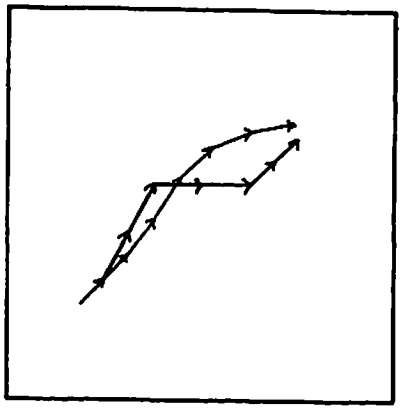


Figure 5.5

This shows the calculated motion from the cross-correlation technique when previous estimates are used. The estimate provided for a given image pair was calculated as the average of the estimates of the previous three image pairs when they exist. The path resulting from the individual estimates is shown as path a. Averaging three consecutive estimates produces path b. Refer to Figure 5.4a for the actual path of the stimulus.

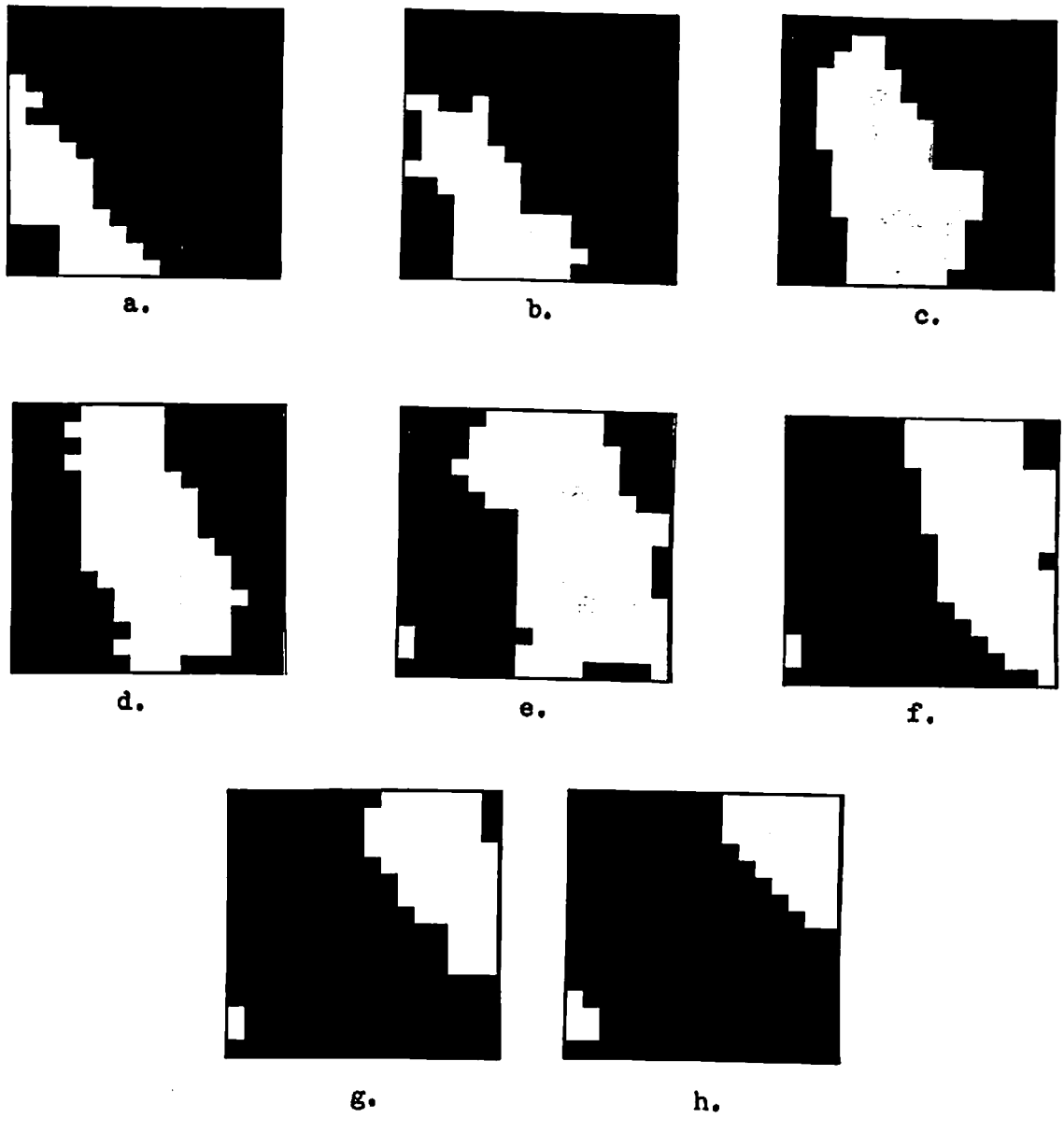
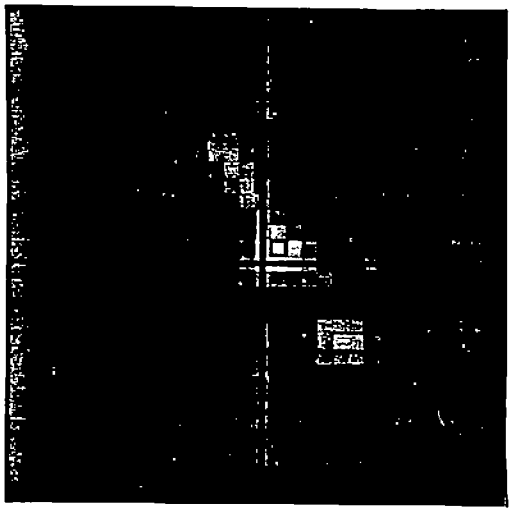


Figure 5.6

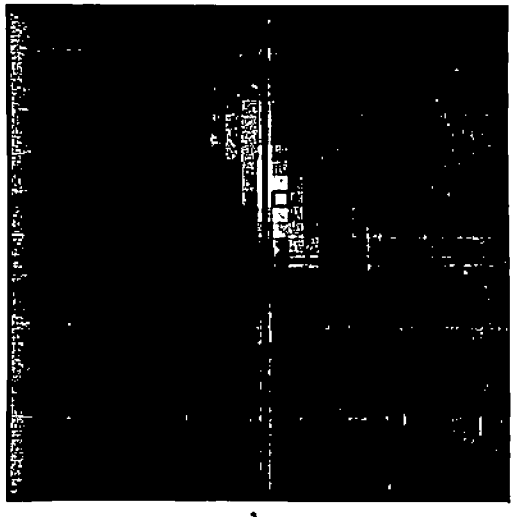
This figure contains the images produced for the masks used to restrict the correlation process in E2. Each element above the threshold is shown in white. These images were produced by thresholding the images in Figure 5.2 at a value corresponding to 25% of the full scale value for that image.

The cross-correlation value was calculated for the intersection areas of the mask images with the values used in the calculation coming from the original images. The resulting cross-correlation images are shown in Figure 5.7a-g. Again the peak in the correlation is depicted with a black border and is an indication of the movement of the object. If the maximum value of the correlation was below 0.5, no motion estimate was retrieved. A broken line is used to border peaks of less than 0.5.

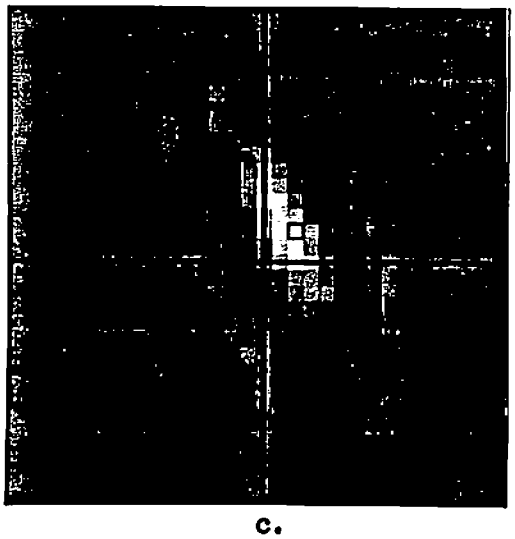
The 0.5 threshold is an arbitrarily chosen level used to differentiate between a match between two images and the case where two consecutive images contain little common structure. (The peaks in the correlation images produced in E1 were all equal to or above 0.75.) This threshold could easily be replaced by a mechanism wherein the magnitude of the correlation peak is used as a confidence indicator. Notice that the ridges of the positive correlation as well as the troughs of negative correlation are much smaller than those in Figure 5.4. Figure 5.8 contains the trace produced by this technique. The movement vectors are given for the first four image pairs as solid vectors indicating that a good match was found as determined by the 0.5 correlation value threshold. The final three estimates are shown with dashed vectors to indicate that the estimates were produced from low correlation values. The motion estimates produced by this technique are no better than the earlier estimates due to the low correlation values found in the final three images.



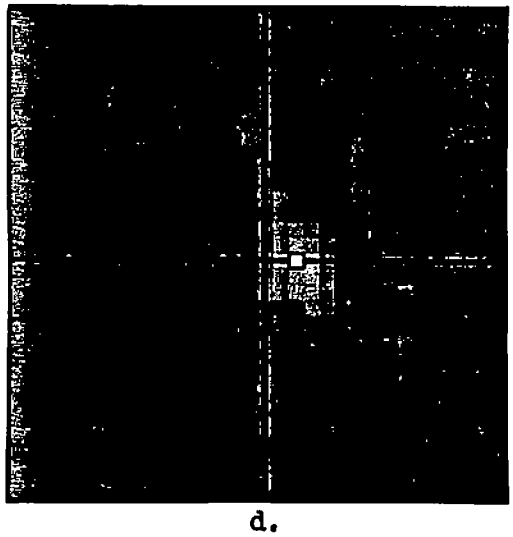
a.



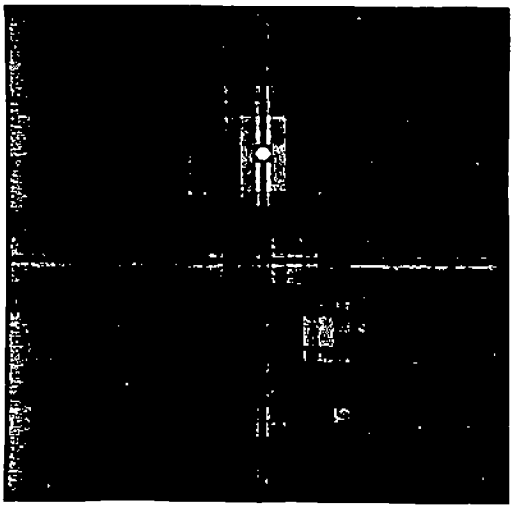
b.



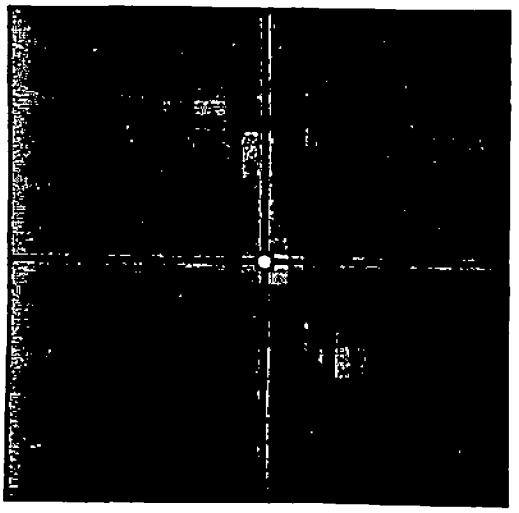
c.



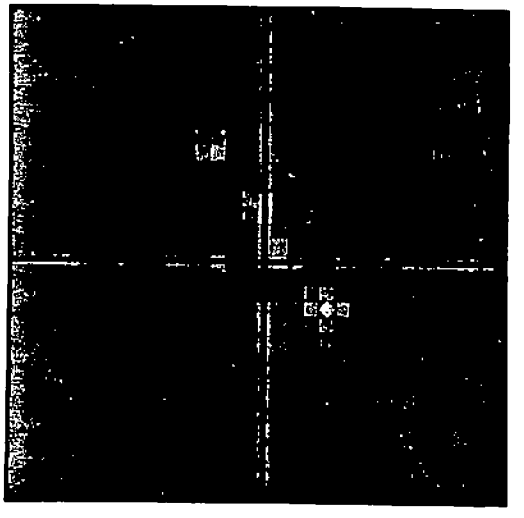
d.



e.



f.



g.

Figure 5.7

These images are the result of the mask-limited correlation process of E2. The binary mask images in Figure 5.6 were used to limit the correlation process to consider only the areas of the original images which were above the threshold. The offset corresponding to the peak in a correlation image is used as the motion estimate. The peak in each image is denoted by the box around the element. A broken line is used for peaks less than a threshold of 0.5, which was arbitrarily chosen, and no motion estimate is provided for those image pairs.

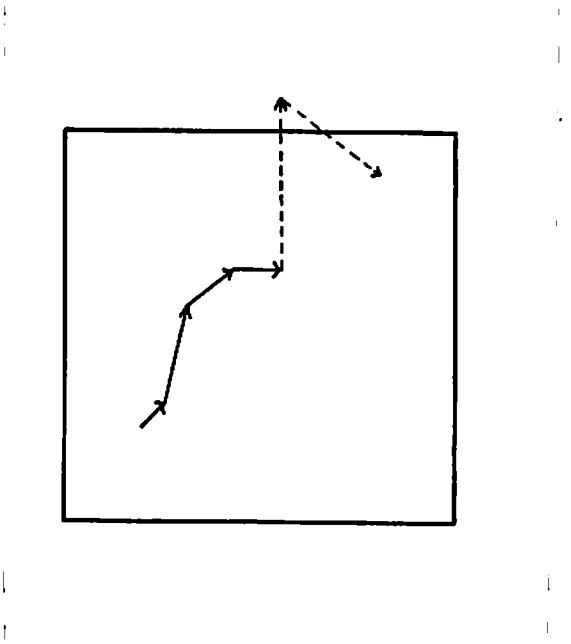


Figure 5.8

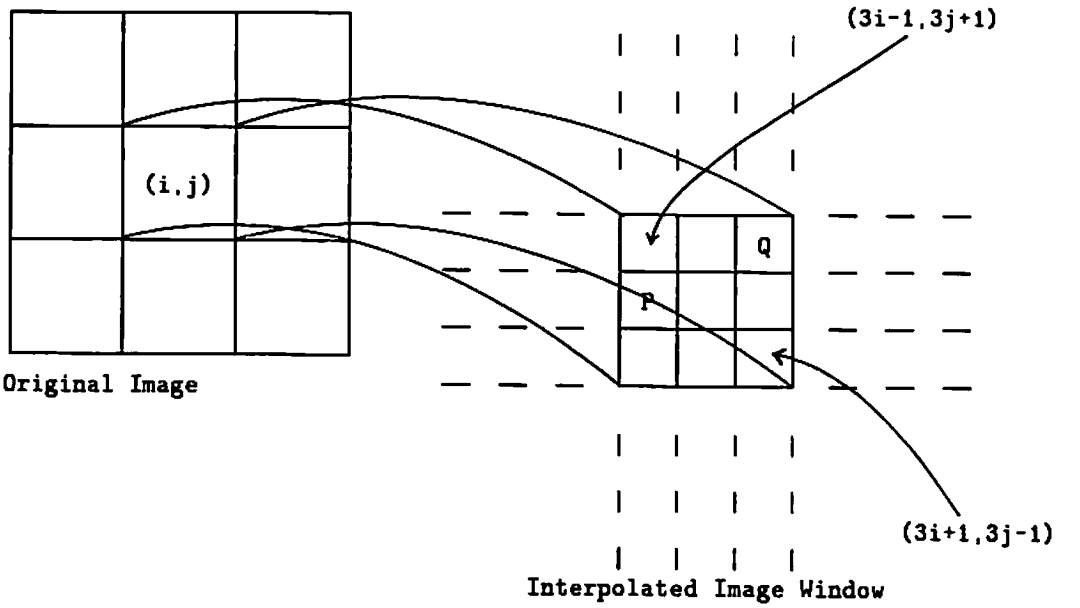
This contains the trace produced by the motion estimates from E2. The offset vectors were obtained from the locations of the peaks in the mask-limited correlation images in Figure 5.7. The dashed vectors represent estimates from correlation peaks of less than 0.5. Note: the final motion estimate was (0,0) thus no vector is visible.

5.3.1.3 E3: Cross-Correlation on Interpolated Data

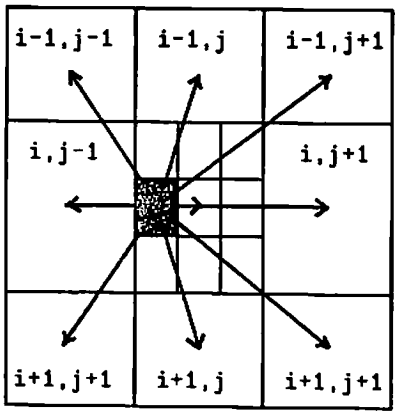
A major problem with the technique of E1 is the fact that the motion vector is limited to an integral number of forcels. This experiment investigates the same technique applied, instead, to images in which sub-forcel interpolation has been used to produce higher spatial resolution images. For each forcel in the original image, a 3x3 window of interpolated values was calculated. This resulted in a second set of images each with nine times the number of forcels. Linear distance based upon distance was used. The value in the interpolated image was calculated as the weighted average of the intensity values in the 3x3 window centered at the corresponding forcel in the original image. This process is depicted in Figure 5.9 for one of the elements in the 3x3 interpolated window.

Figure 5.9

This diagram shows the process by which an interpolated image is produced from an original image. 5.9a: shows the registration of the 3x3 interpolated window with the original window. The coordinate system for the interpolated image was chosen simply to ease implementation. 5.9b illustrates the distance weighting process. The distances are calculated from the center of the forcel whose value is being calculated when registered with the original window to the center of the original forcel of interest. 5.9c contains the mask weights for interpolated forcel q in 5.9a. 5.9d contains the mask weights for interpolated forcel p in 5.9a.



a.



Distance Vectors

b.

0.0831	0.0948	0.0599
0.1498	0.2997	0.0749
0.0831	0.0948	0.0599

Mask Values for Element P

c.

0.0763	0.1448	0.1145
0.0785	0.2290	0.1448
0.0572	0.0785	0.0763

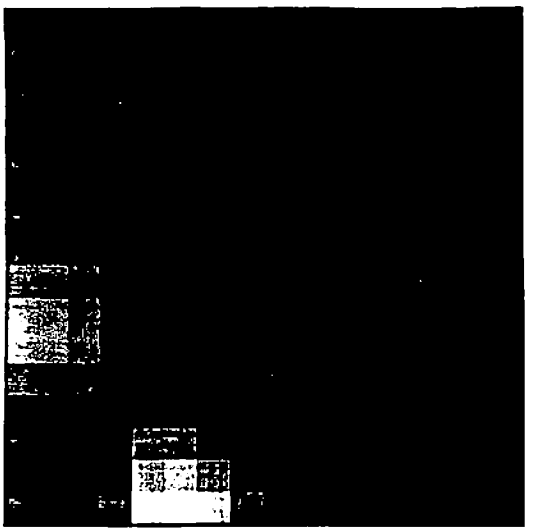
Mask Values for Element Q

d.

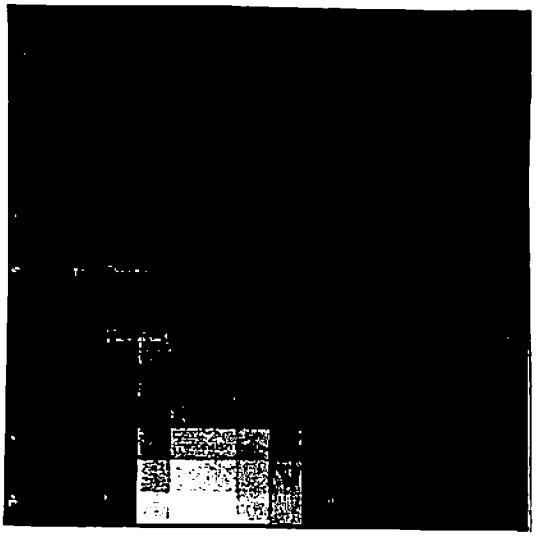
The values of the weights are the Euclidean distances measured from the center of the forcel of interest to the centers of the forcels being considered normalized by the sum of the distances. The distance of the center forcel in the interpolated window to the center forcel in the original image is defined as 1 instead of the actual distance of 0. Applying this interpolation process to the images in Figure 5.2 results in the set of interpolated images shown in Figure 5.10.

These interpolated images were then used as the input to the cross-correlation process of the first experiment. The resulting cross-correlation images are shown in Figure 5.11. The images contain the same global structure as those in Figure 5.3 but appear much smoother due to the higher spatial resolution.

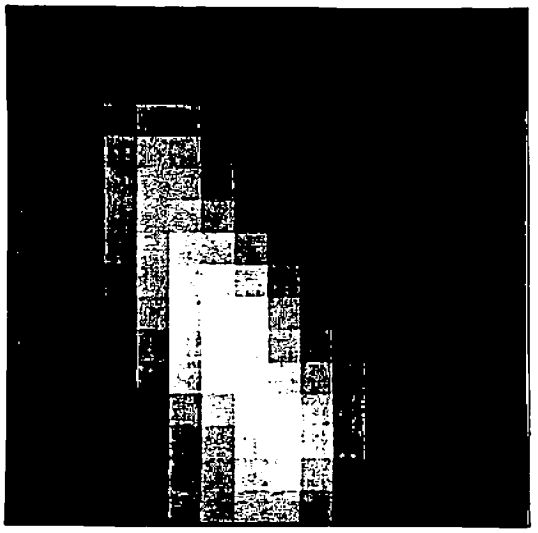
The motion calculated from these images is shown in Figure 5.12. The angle of the path calculated from the average of the motion vectors is 0.71 radians with a speed of 1.59 forcels per image compared with the actual values of 0.46 radians and 1.79 forcels per image. The error for this technique is slightly larger for the angle of the path but less for the speed of movement, thus indicating that linear interpolation does not provide much improvement in the average motion calculations. However, comparisons of the "a" tracks in Figures 5.4 and 5.12 predicted from the calculations show that the placement of the center of the final cylinder is closer to the actual location when the interpolated data are used.



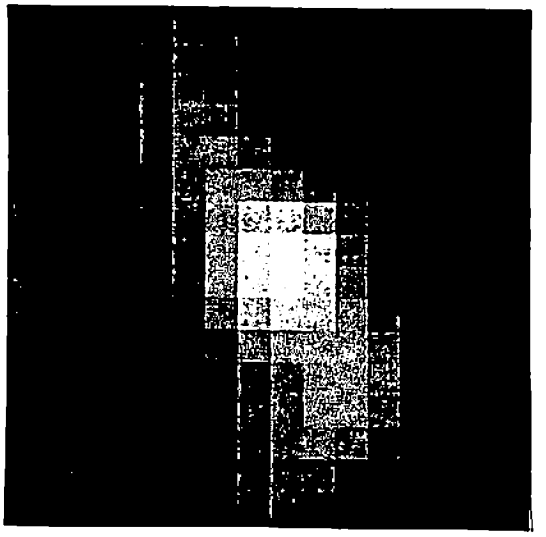
a.



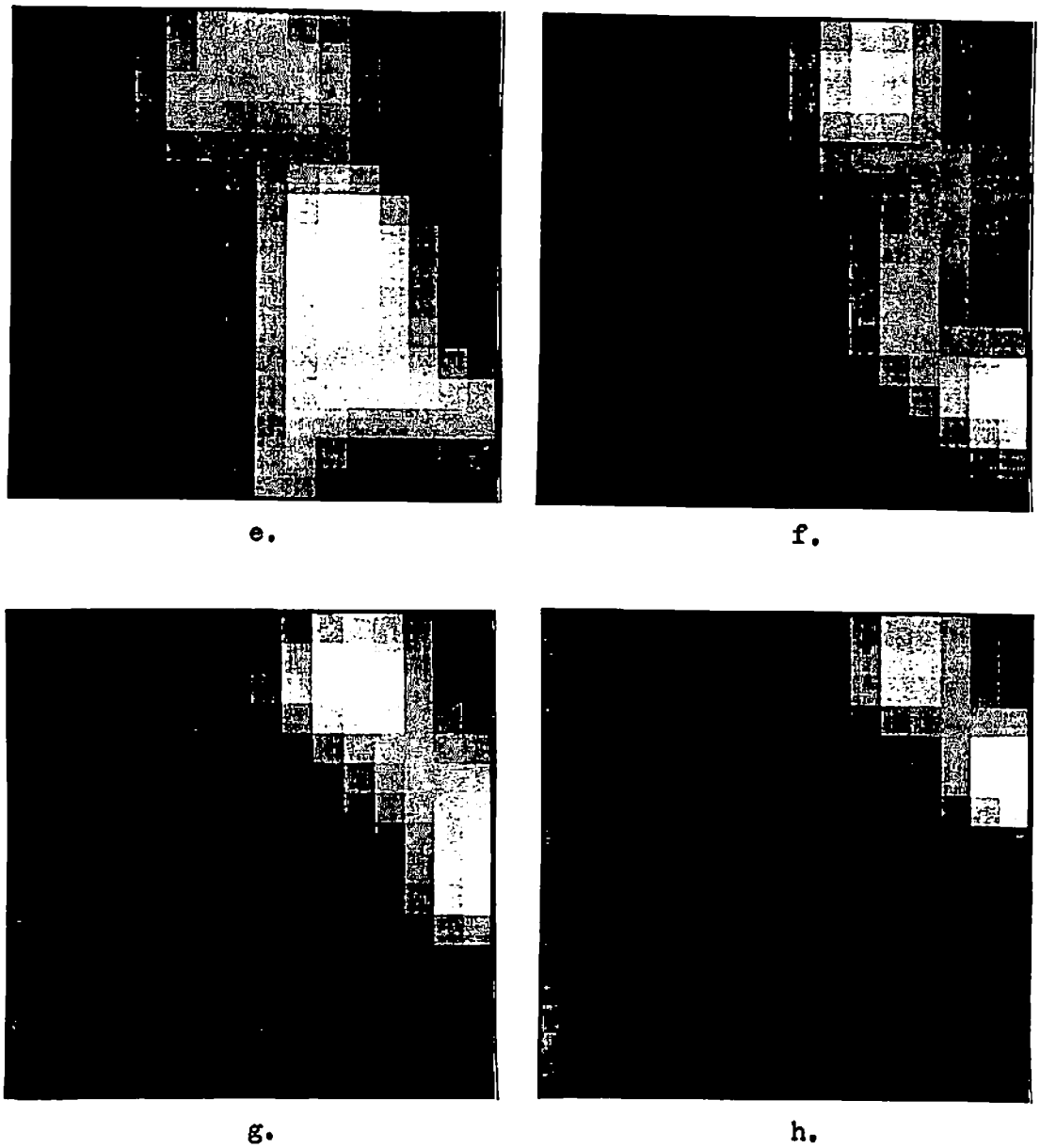
b.



c.



d.

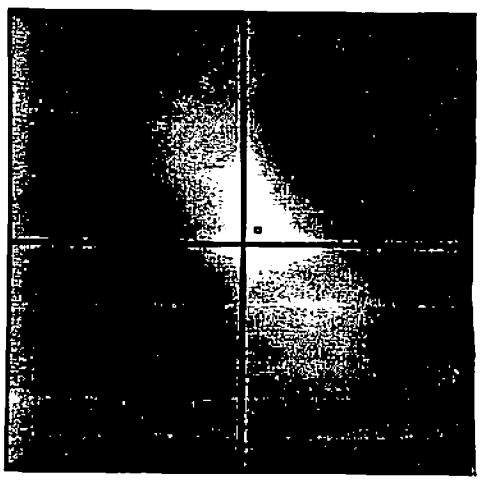


e. f.

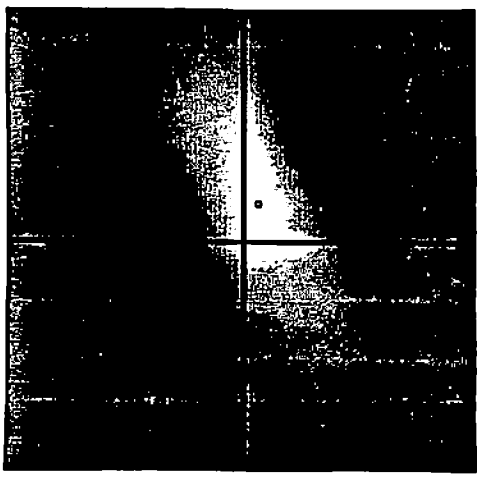
g. h.

Figure 5.10

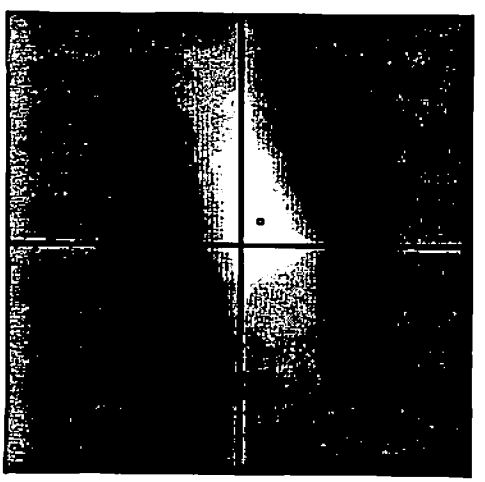
This contains the set of images interpolated from the images in Figure 5.2. Each forcel in these images was calculated as the weighted average of the forcels in a 3x3 window in the original image centered at the original forcel containing the forcel being updated. The weights are based on the scheme depicted in Figure 5.9.



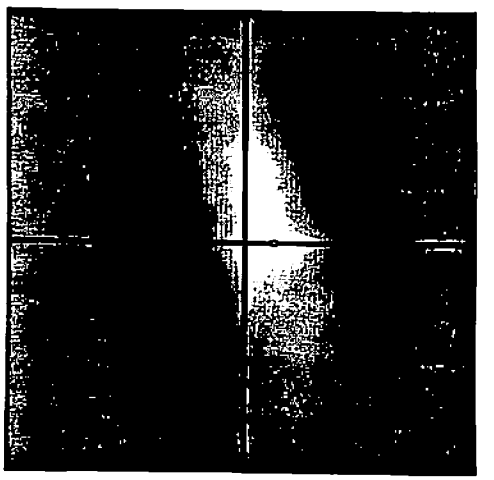
a.



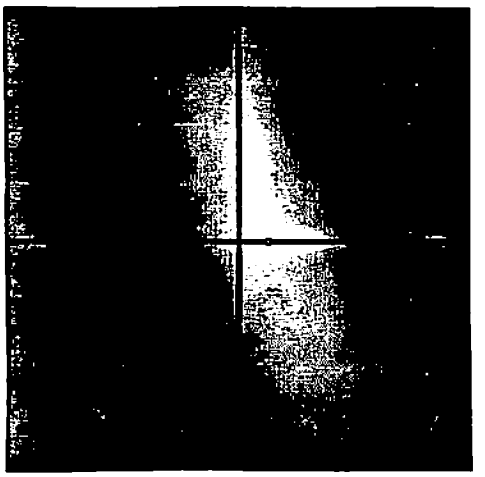
b.



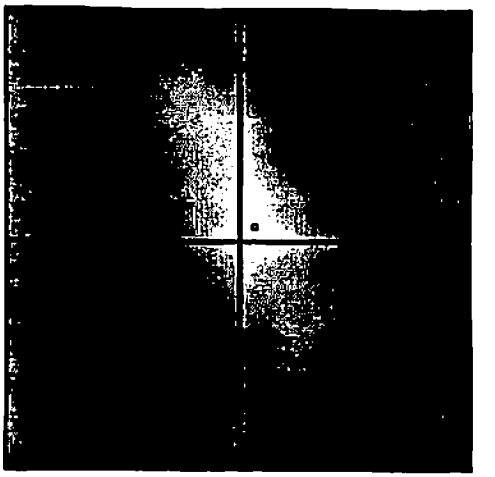
c.



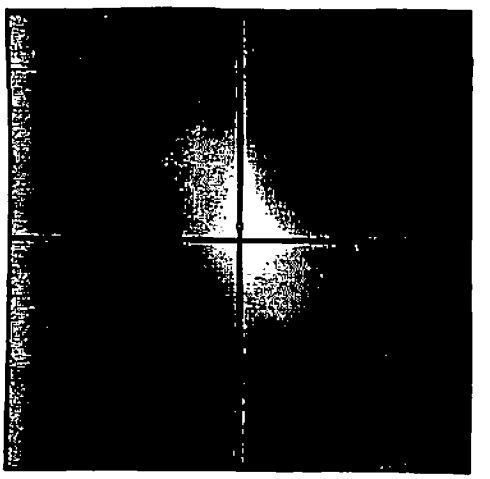
d.



e.



f.



g.

Figure 5.11

This figure contains the cross-correlation images produced from the interpolated data images in Figure 5.10. The images contain the same global structure as those in Figure 5.3 but appear much smoother due to the higher spatial resolution.

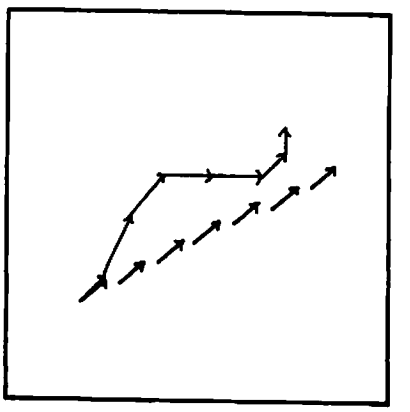


Figure 5.12

The motion calculated from the correlation images shown in Figure 5.11 is contained here. The angle of the path calculated from the average of the motion vectors is 0.71 radians with a speed of 1.59 forcels per image compared with the actual values of 0.46 radians and 1.79 forcels per image. As in the earlier figures, track a results from placing the calculated offset vectors head to tail and track b from placing an average offset vector at the actual locations.

As with the basic cross-correlation technique, the estimates from the processing of one image pair can be used to predict the motion in subsequent pairs. The technique used to generate the trace in Figure 5.8 was used in this experiment with interpolated data. The resulting trace appears in Figure 5.13. As before, the motion estimates for three consecutive image pairs were averaged to produce the prediction used for the next image pair. The cross-correlation values were weighted by their distance from the predicted offset vector. The distance weight was a linearly decreasing function with a radius of twice the sensor width. From the figure, it is clear that the motion as estimated by this procedure is closer to the actual motion than any of the other estimates, thus illustrating that it is possible to use previous estimates to improve later estimates.

5.3.2 Motion: Feature-Matching Techniques

Experiments E4 through E9 represent feature-matching techniques. The experiments range from the tracking of simple unitary features of the images to the matching of dense fields of features. Such techniques rely on the processing of the raw data images to extract features and then the tracking of these fea-

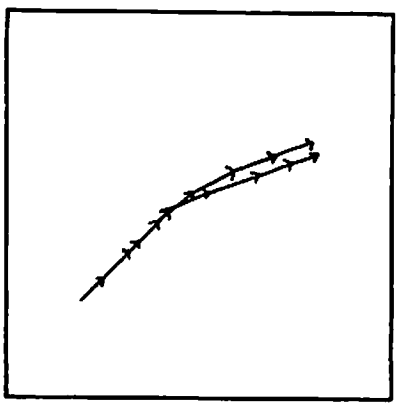


Figure 5.13

This contains the trace produced using the interpolated data and previous estimates to bias the selection process. The results here are clearly superior to those produced by the earlier techniques. Track a results from placing the calculated offset vectors head-to-tail and track b from placing an average offset vector at the actual locations.

tures over time. The tracking of features is essentially the matching problem discussed at length in the machine vision literature. That is, if a set of features is obtained for each of two images, visual or tactile, separated in time, how are the features in one image matched to those in the other image? The following set of experiments presents the results of some feature-based techniques for extracting tactile motion information.

5.3.2.1 E4: Binary Region Centroid Tracking

The highest level feature one could extract from a tactile image is a parametric description of the object being sensed, including the parameters of the object's position. Since the tactile sense is a contact sense, any non-zero elements in a force image represent points of contact with the object. Simple thresholding of a tactile image results in two types of regions: those below threshold representing areas wherein there is little or no contact with an object and those above the threshold which represent areas of contact. If it is assumed that the sensor is contacting only one object at any given time, then the above-threshold regions are indicators of the object. A simple measure of the above-threshold regions can be used as an estimate of the object's position on the sensor.

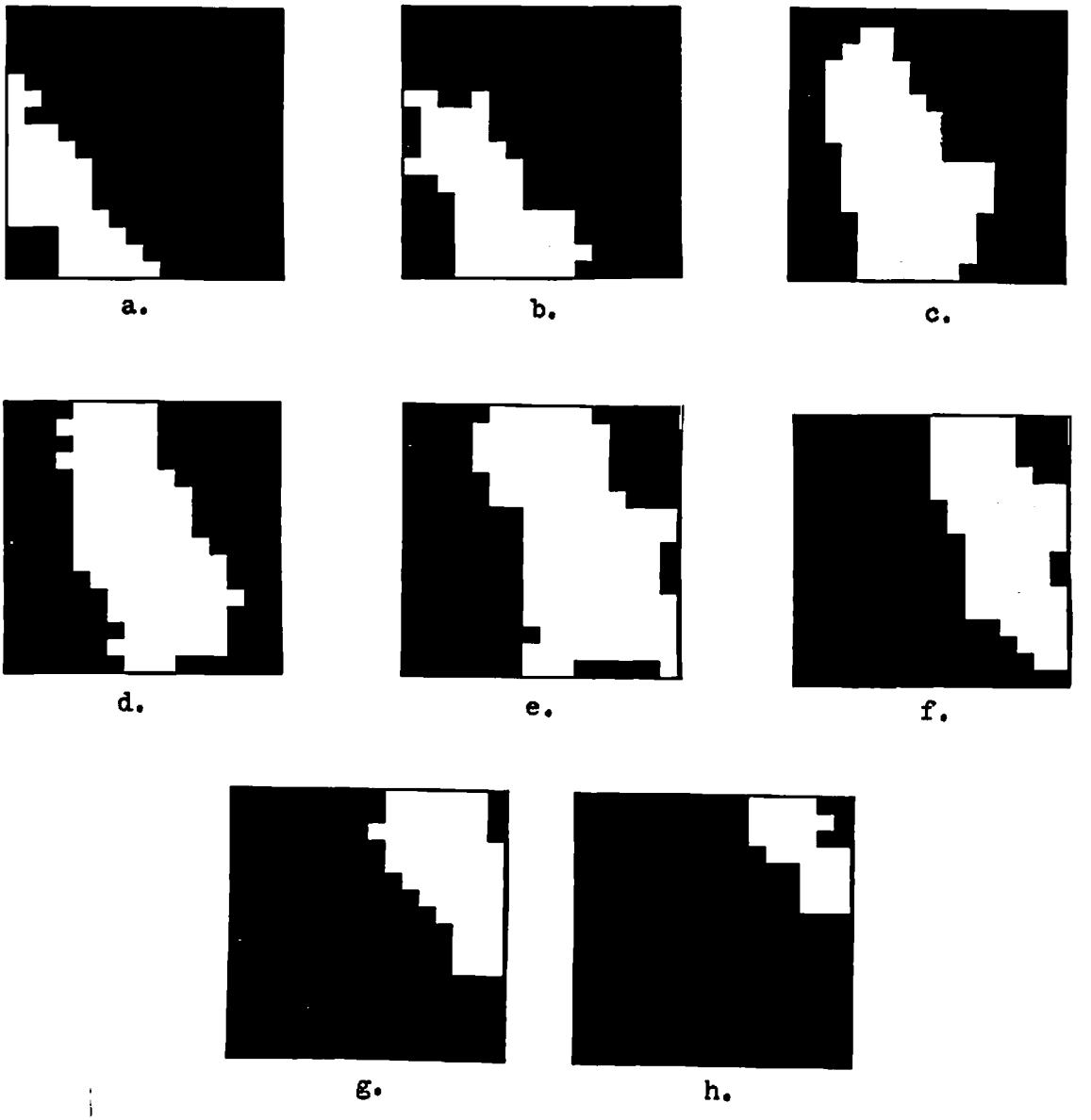


Figure 5.14

These are the binary images generated from the images in Figure 5.2 for experiment E4. A threshold of 25% of the range of the data was used to produce the images in 5.14a, b, c, and d. These first four images are the same as the binary images in Figure 5.6a, b, c, and d. 5.14e and g resulted from a threshold of 30%, 5.14f from 35%, and 5.14h from a threshold of 45%. Forcels above the threshold are displayed in white.

Figure 5.14 contains the eight binary images corresponding to the moving cylinder images in Figure 5.2. These images were produced by thresholding each of the original images at the smallest threshold, of at least 25%, which resulted in only one region above the threshold. The actual thresholds used are provided in the caption of the figure. The centroid of the region above the threshold was found for each image. The set of centroids comprise a sequence of estimates for the object's location. These estimates are plotted in Figure 5.16.

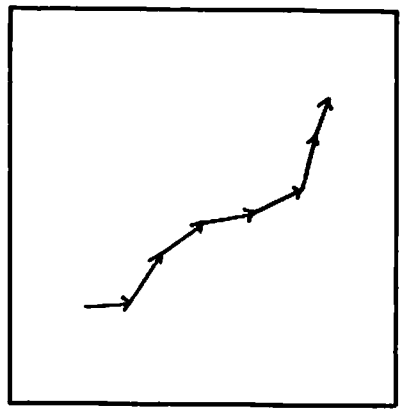


Figure 5.15

This trace shows the motion estimate produced by tracking the centroids of the regions in Figure 5.14. Average offset is reasonably close to the actual value. However, this is misleading since the centroid is not an accurate estimate of a parameter fixed relative to the object coordinate system. For instance, the centroids seen in the last two frames are in roughly the correct location, cf. figure 5.4a, but the object is clearly off the sensor and thus the centroid of the region may not map to the same location on the object as in 5.15d.

The mean centroid displacement vector between consecutive images is (0.09,0.07) compared with the actual displacement of (0.1,0.05) which indicates that the trend is correct but the absolute displacements are slightly low. While the estimate is reasonable, this is misleading since the centroid is not an accurate estimate of a parameter fixed relative to the object coordinate system. For instance, the centroids seen in the last two frames are in roughly the correct location, cf. figure 5.4a, but the object is clearly off the sensor and thus the centroid of the region may not map to the same location on the object as in

5.15d. This is a crude estimate in that it may indicate movement of a feature not directly related to the object, but the information may be useful given the proper task.

5.3.2.2 E5: Ridge Estimator Tracking

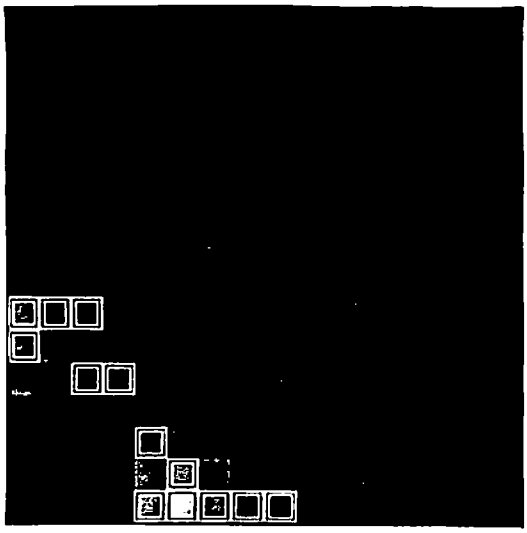
In this experiment a ridge estimation algorithm from Chapter 4 was applied to each of the eight cylinder images. The algorithm consisted of calculating the regression line for a set of points chosen from the image. The line is accepted as an estimate of the ridge if the value of the Pearson correlation coefficient is greater than 0.6. The points are chosen by scanning the rows and columns and choosing the forcel with the maximum value in the row or column if it is greater than some threshold. In this case the threshold used was the value corresponding to 50% of full scale.

Figure 5.16 contains the selected points and the corresponding ridge estimators. The images are displayed in pairs with the left image showing the selected feature points and the right image the ridge estimator derived from those feature points. No ridge estimator was calculated for the features selected in the final four images in the sequence since the correlation coefficient was below 0.5. This is an arbitrarily chosen threshold meant to differentiate between strong and weak fits of a line to the selected features. The blank images to the right of these images are meant to signal that no estimator was calculated. The average motion vector calculated between the first four images is (0.12,0.07). Unfortunately, the lack of estimates for the final four images results in too few data points for this technique to be evaluated. However, it is the purpose of this work to demonstrate the approaches and uses of tactile sensor data and even the failures provide insight.

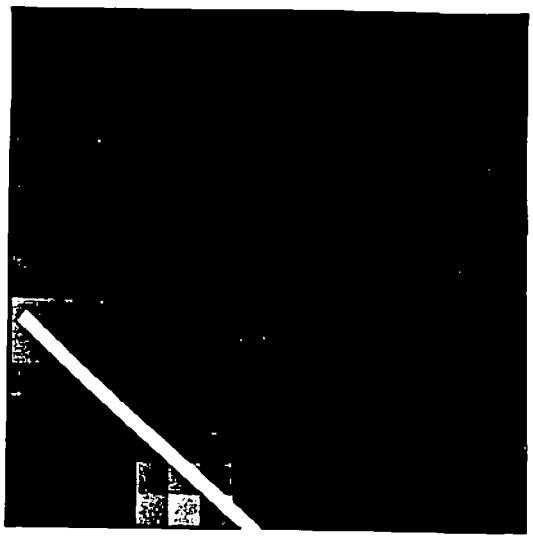
5.3.2.3 E6: Hole Center Tracking

This experiment extends the hole-finding algorithm discussed in Chapter 4, E3, to a sequence of images. For each image in a set of five of a plate with a hole moving across the sensor, the hole-finding algorithm was used to estimate the location of the center of the hole. Figure 5.17 contains the individual estimates and Figure 5.17f contains the last image in the sequence with all of the estimates superimposed. Note that the images used contain 100 forcels since they were collected from an earlier version of the tactile array sensor. The data images are unpreprocessed except for the replacement of values from bad forcels by the average of the 3x3 neighborhood about the forcel. Base level offsets and response range scaling has not been performed on these images. The data images were created under uncontrolled circumstances so that the actual motion is not known a priori.

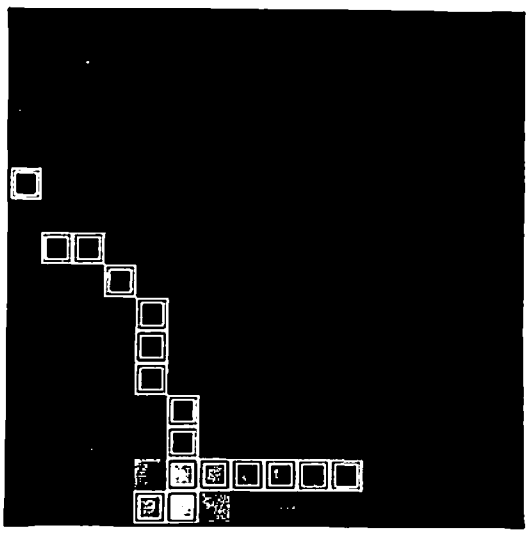
In this case the algorithm was applied to each image with no knowledge of the results from processing of the previous image. As has been demonstrated for E1 and E3, the estimate derived from earlier pairs of images could be used to predict where the algorithm should concentrate its search in subsequent pairs. Information thus derived could be used to build a model of the object's



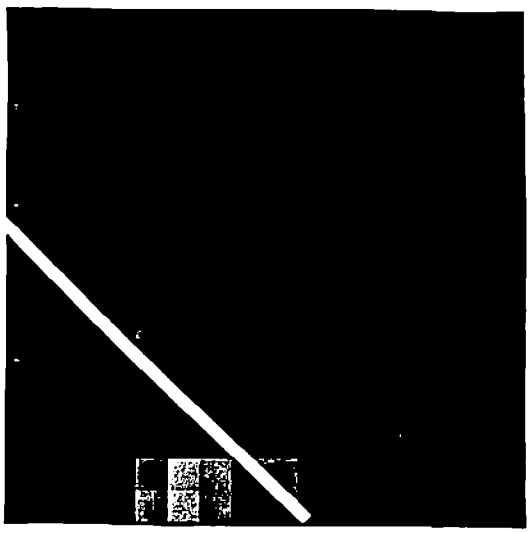
a.



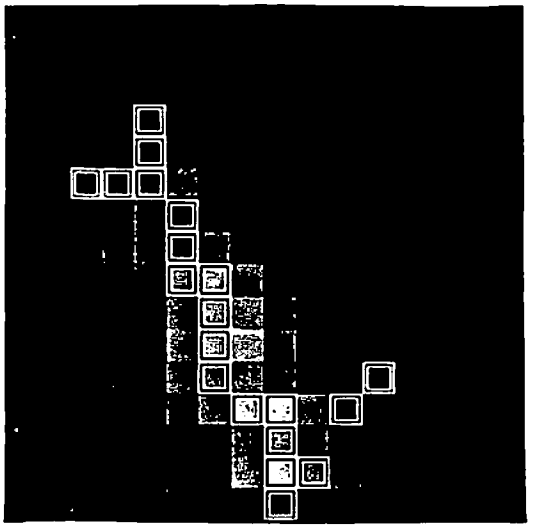
b.



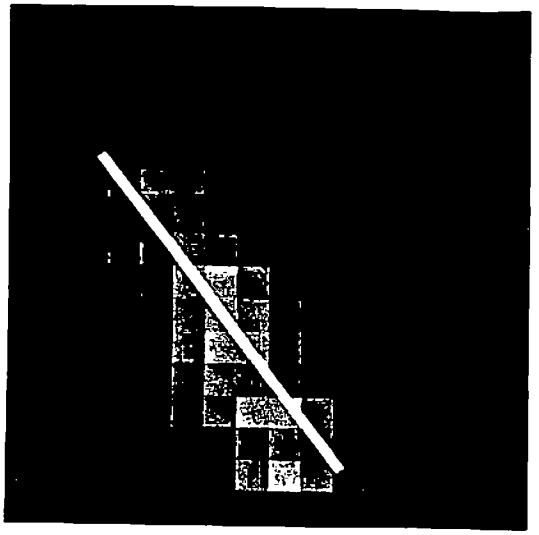
c.



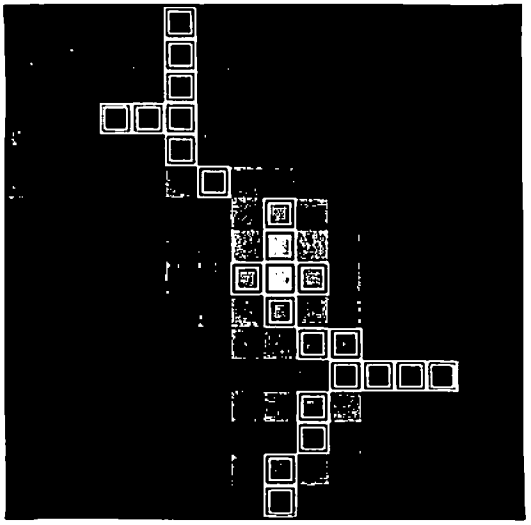
d.



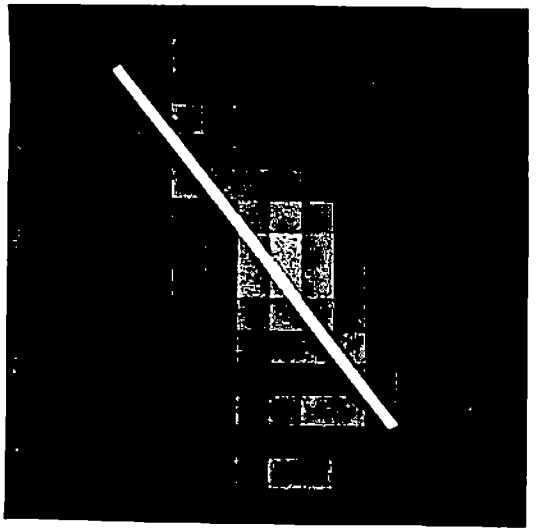
e.



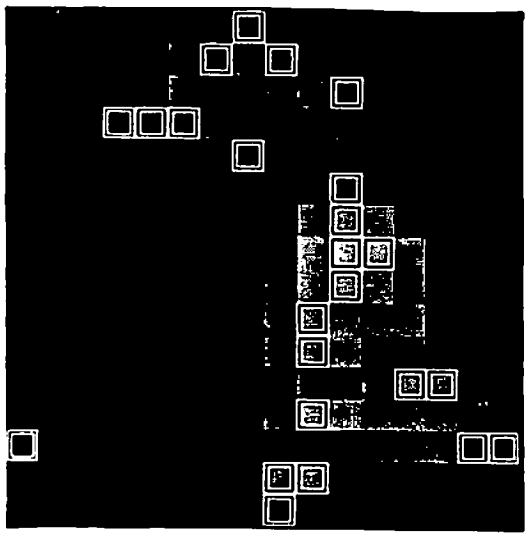
f.



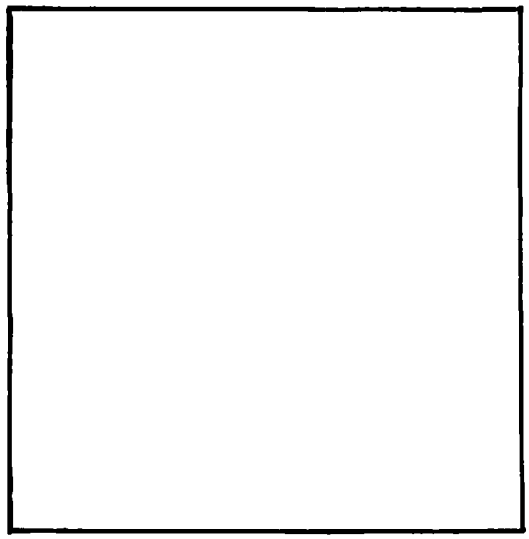
g.



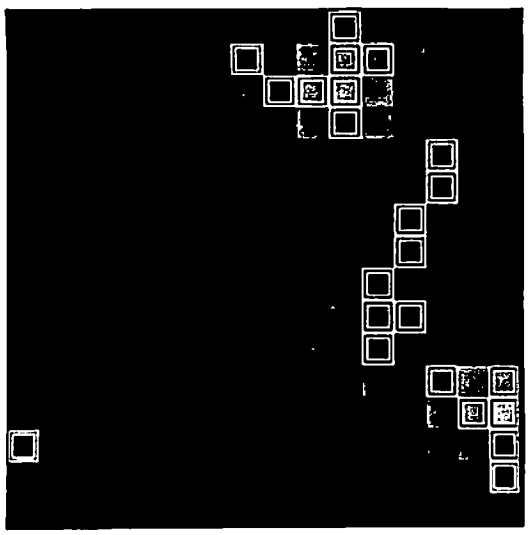
h.



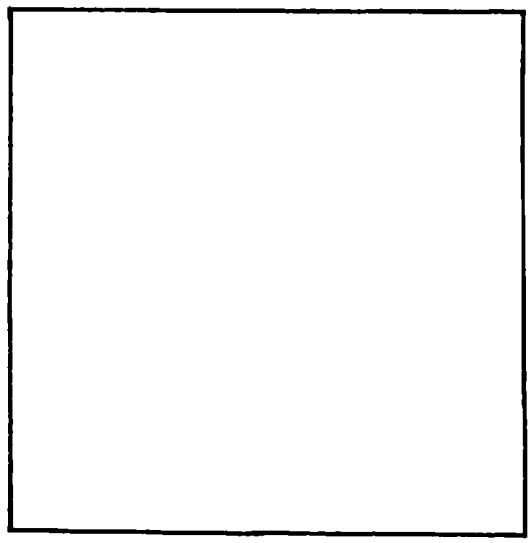
i.



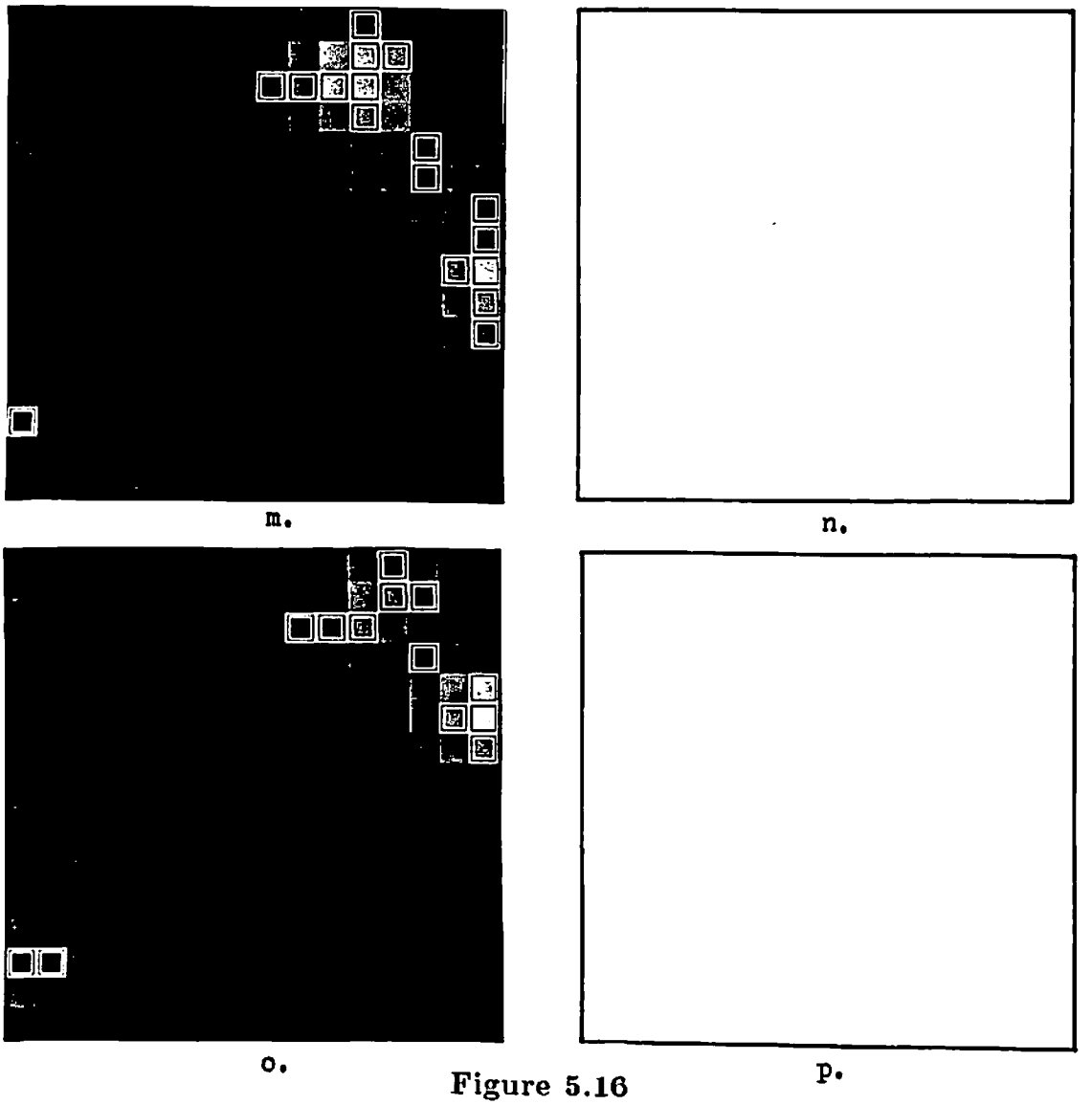
j.



k.



l.



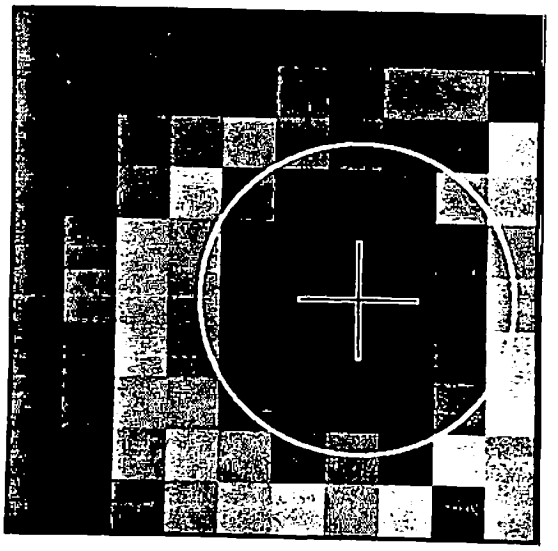
m. n. o. p.

Figure 5.16

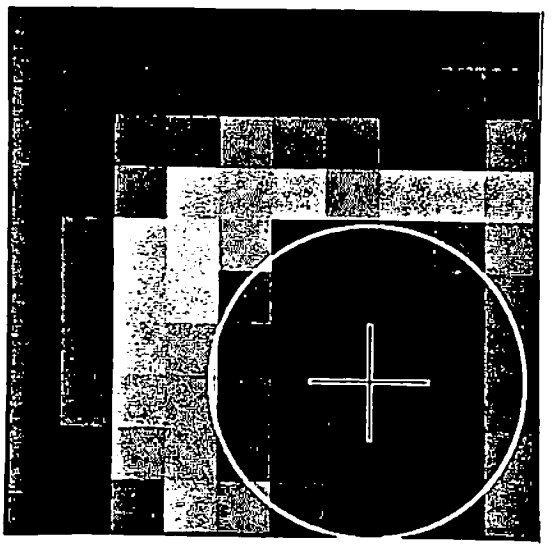
This figure contains the selected points and the corresponding ridge estimators. The images are displayed in pairs with the left image showing the selected feature points and the right image the ridge estimator derived from those feature points. No ridge estimator was calculated for the features selected in the final four images in the sequence since the correlation coefficient was below 0.5. The blank images to the right of these images are meant to signal that no estimator was calculated. The average motion vector calculated between the first four images is (0.12.0.07).

Figure 5.17

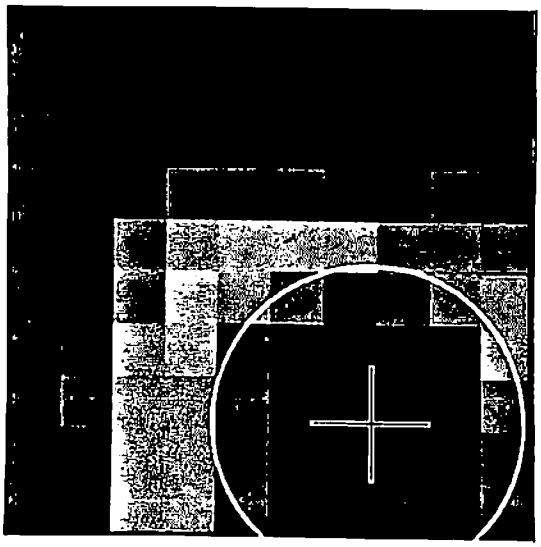
This contains the results of the hole-finding algorithm extended to handle a sequence of images. The images in 5.17a-e are the original images with the hole location estimates overlaid. The images were collected from the Version 1 sensor and contain a 10x10 array of forcels. 5.17f shows the last image in the sequence with all five estimates displayed. The trend is correct; however, a lack of knowledge regarding the actual locations makes further evaluation impossible.



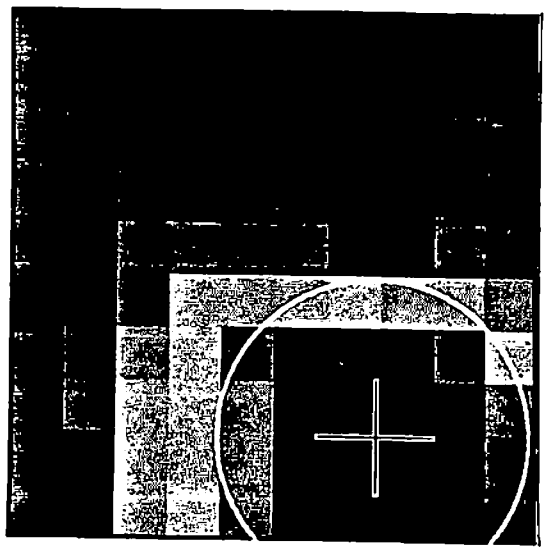
a.



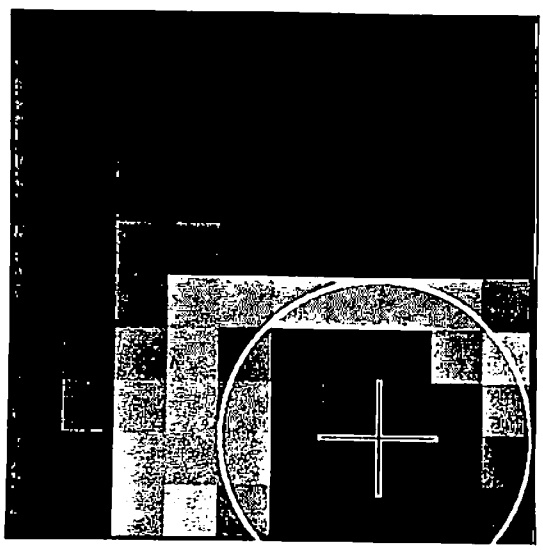
b.



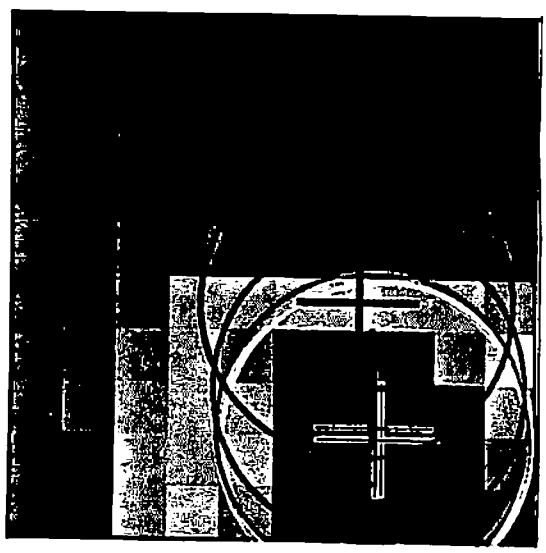
c.



d.



e.



f.

motion and this model used to reduce the search space in succeeding applications of the algorithm.

5.3.2.4 E7: Edges

The previous three experiments relied on the extraction of high-level features of the object being scanned. The first used the centroid of the contact pattern, the second an estimator of the location and extent of a ridge, and the third an estimate of the center of a hole. Experiment E7 relies on a lower level feature, extending the edge extraction process discussed in Chapter 4. Briefly, a one-by-two difference operator is applied to an image as the first processing step. This results in an image containing the difference, both horizontal and vertical, between the forcels. These differences correspond to edges and provide the input to a probabilistic relaxation procedure [89] which results in an image containing the probabilities of edges between each pair of forcels. The images shown in Figure 5.18 are the results of applying 10 iterations of the relaxation procedure to the edges from the images in Figure 5.2.

Notice that the edges resulting from this procedure represent the boundaries in the tactile image to a reasonable degree. The long connected edges tend to correspond to the edges of the cylinder's image and the shorter edges to image areas not associated with an actual edge. The edges produced in Figure 5.18c are especially clear. Notice, however, that the edges of 5.18f are not clear, and there is confusion in the lower left corner. From this point, the edge images are processed to fit long boundaries to the edge chains seen in the figure. These boundaries can be matched through time using parameters of length, orientation, location, and previously calculated motion to approximate the movement of the cylinder.

5.3.2.5 E8: Difference Image Edges

The extraction and thinning of edges through relaxation can also be applied to the results of a previous processing step. In this case, difference images between consecutive image pairs were processed to extract the edges. This technique is derived from the visual image processing domain where differencing has been used for tracking, object extraction, and motion detection, cf. [105]. It is known from the characteristics of the domain that scaling and perspective distortions are not a problem. It is assumed that there is only one object contacting the sensor at a given time and that the same object is being sensed in images adjacent in time. If no change in the position of an object occurs between the times that two images are collected, then there should be no differences between the images, i.e. forcel by forcel subtracting image n from image n+1 should yield an image of zeros. Conversely, if the object undergoes a positional change between the two images, the difference between those images will have peaks and troughs corresponding to the movement of the object. It should be noted that the surface texture of an object will affect the difference image if the texture is large enough to be distinguished by the sensor. The

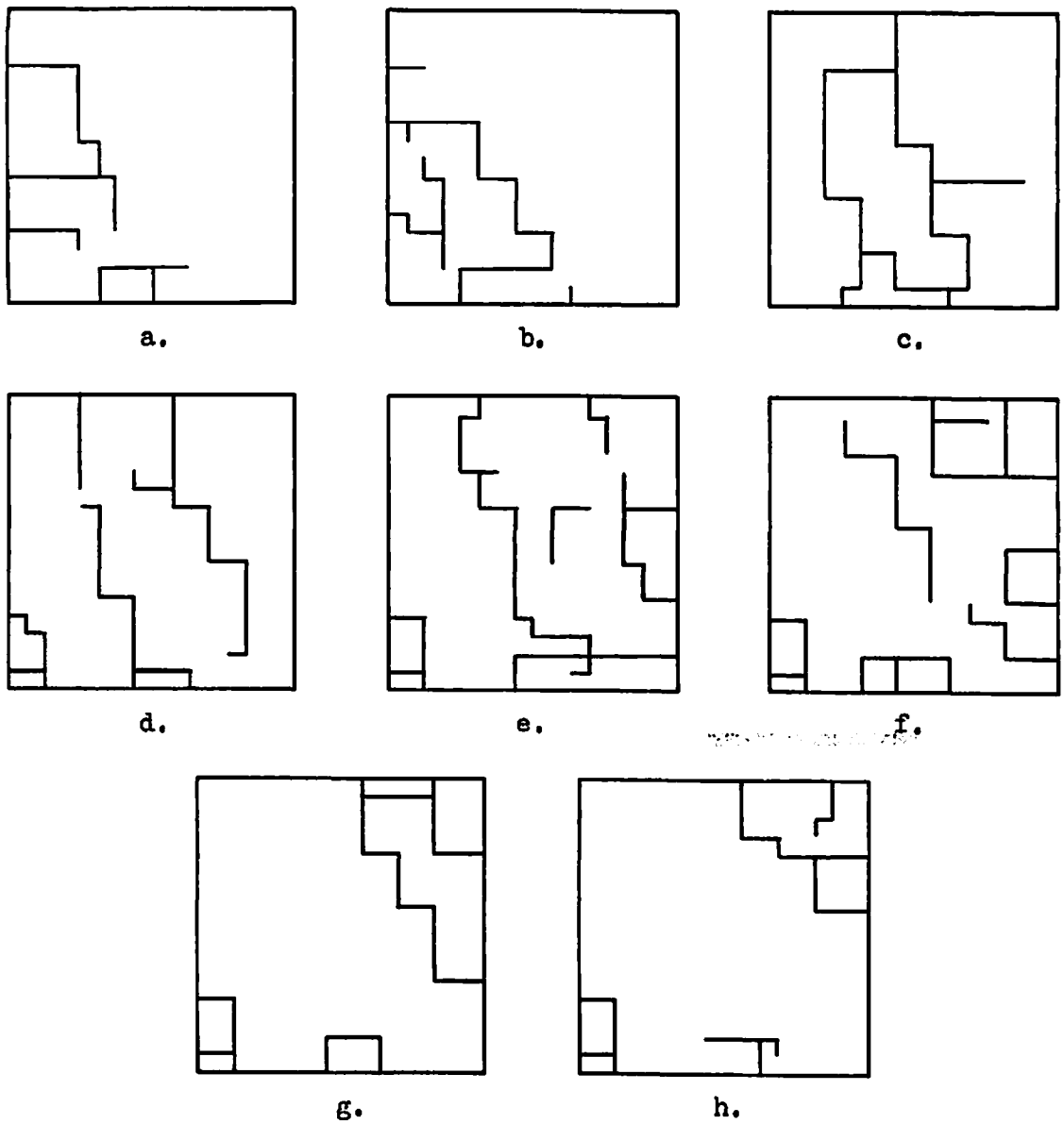


Figure 5.18

This figure contains the results of an edge relaxation procedure. The initial images were those shown in figure 5.2. A 1x2 difference operator was used to generate initial estimates of the vertical and horizontal edges in each of these images. These data were then supplied to a relaxation procedure which was allowed to iterate 10 times and produced the images contained here.

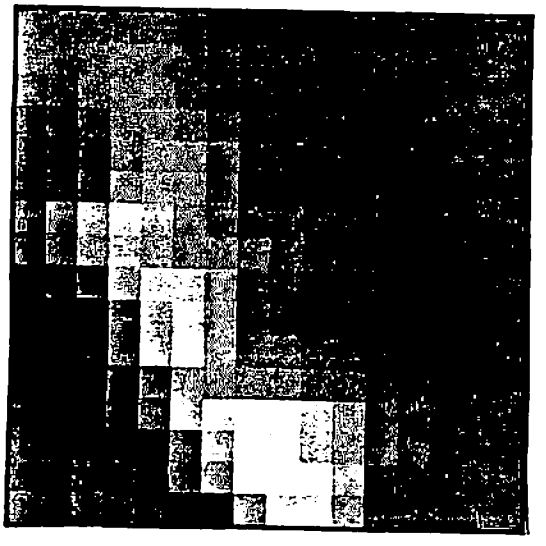
peaks and troughs mentioned here are considered to be global features of the structure of the difference images.

Recall that the forces present on the sensor surface caused by the interaction with an object produce positive numbers whose magnitudes increase with increasing force. If image nth in a sequence is subtracted from image n+1, the positive areas in the resulting difference image correspond to the areas in image n+1st occupied by the object which are not by the object in image nth. That is, the positive areas in the difference image correspond to the leading area of the object. Similarly, the negative areas correspond to the area of the imprint of the object which appears in the nth image and not in image n+1st, i.e. the trailing area of the object. The area in the difference image between the positive and negative areas corresponds to the overlap of the imprints of the object. This analysis rests on the assumption that the stimulus object produces a single region with no voids and that the motion of the object was less than the dimensions of the object.

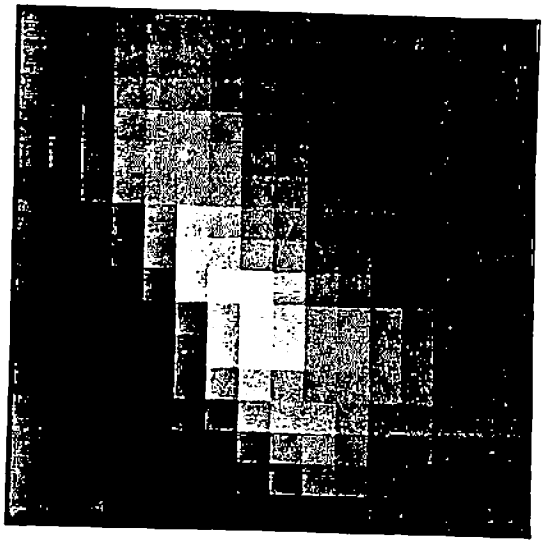
The tactile image of an object may include holes, cf. Figure 5.17, and may exhibit movement between frames which is larger than the object. In the case of a hole, positive areas in the difference image represent leading edges of either the object or possibly a feature of the object if the feature produces positive values in the image. The holes in the images in Figure 5.17 are features of the image but they produce areas of low force in the tactile images. As these regions move and overlap areas which correspond to high force areas in the previous images, troughs appear in the difference image corresponding to their leading edges; thus, the interpretation of the peaks and troughs is actually dependent upon the structure of the images. In the case considered here, the tactile images contain no holes and the simple analysis suffices.

If the motion of the object between two images is greater than the width of the object, then the previous analysis is also incorrect. In this case, the positive area in the difference image corresponds to the entire object in its location in image n+1. If the motion is on the order of the width of one forcel, then the positive and negative peaks appear one forcel wide. If the motion is substantially less than one forcel, the resulting difference image will have very poorly defined leading and trailing edges. The actual motion in this experiment was 1.79 forcels per image pair so that in the case of a perfect transducer the peaks and troughs in the difference image should be almost exactly two forcels wide. The sensor provides some local averaging during the transduction process and additional blurring is added by the image correction process so the ridges appear wider.

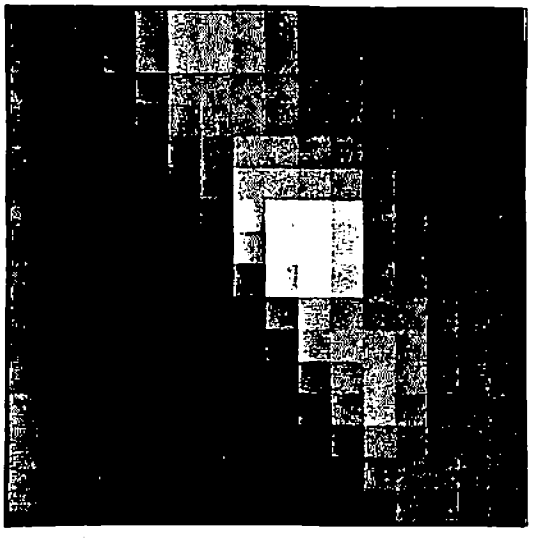
Figure 5.11 contains the seven difference images calculated from the data images in Figure 5.2. The images are ordered from the oldest to the newest. That is, image 5.2a occurred prior to 5.2b, etc. The difference images were calculated as image n+1 minus image n on a forcel-by-forcel basis. Notice that the first four difference images contain relatively sharp peaks and troughs corresponding to the leading and trailing edges of the cylinder. The later



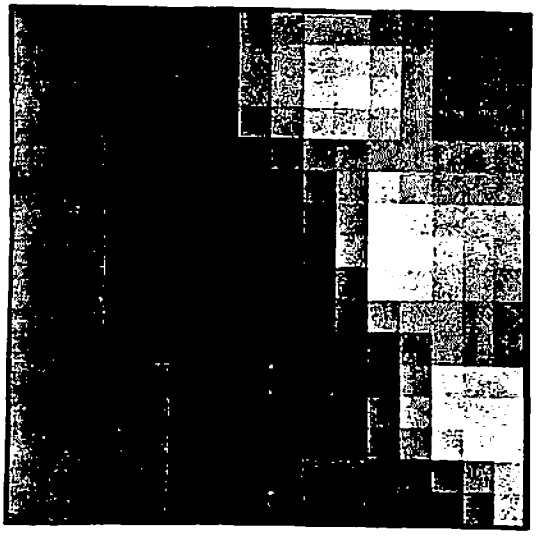
a.



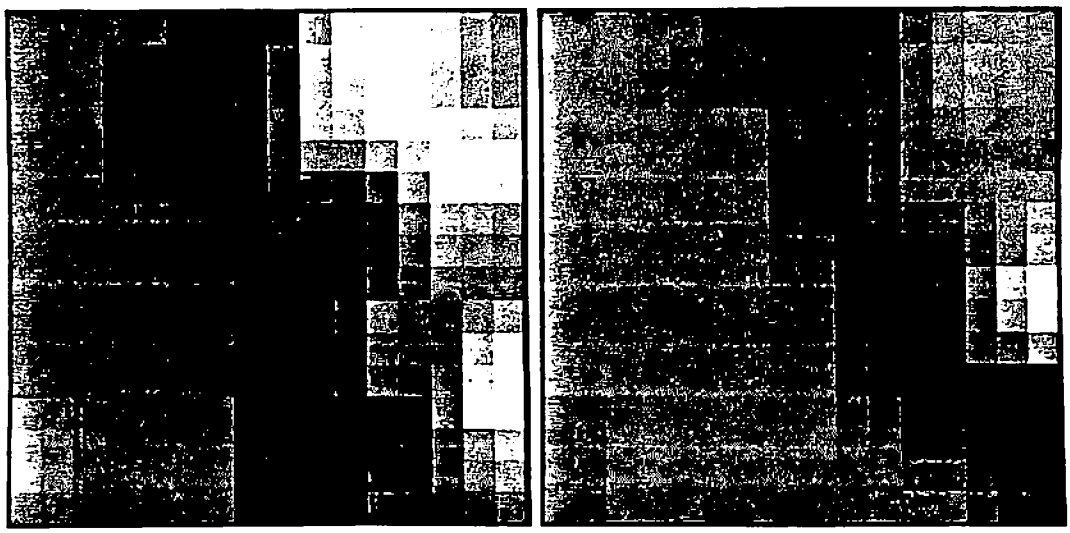
b.



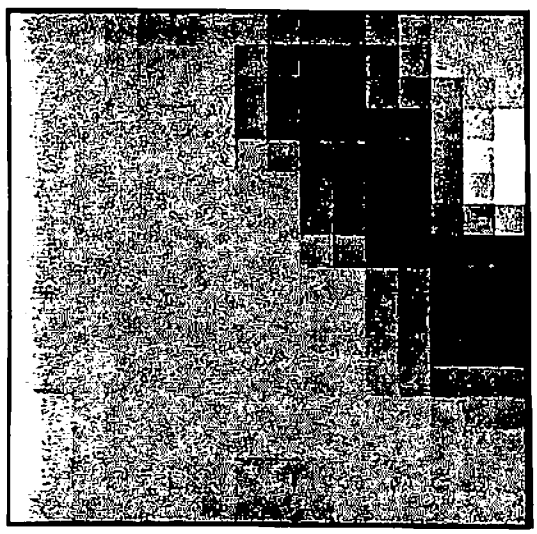
c.



d.



e. f.



g.

Figure 5.19

This figure contains the set of difference images derived from the images in the moving cylinder sequence shown in Figure 5.1. Each image was generated by subtracting image n from image n+1 on a forcel-by-forcel basis.

difference images have poorer definition since the data images contain structure which is less distinctly defined.

Thus a difference image contains an area of high values corresponding to the section of the object in image n+1 which does not appear in image n. This area contains the movement information. If the magnitude of the movement is on the order of the size of a forcel, then the area will appear as a ridge, one side of which indicates the leading edge of the object's imprint. Similarly, there exists a trough which corresponds to the trailing edge of the object. The application of the edge relaxation technique to such images results in long edges in the images which correspond to the leading and trailing edges of the object. Figure 5.20 contains the results of this process.

The difference images were not, for the most part, truly clear. The edge images resulting from the processing of these images also lack clear detail. Many spurious edges appear in the images in addition to the edges matching the leading and trailing edges of the object. In fact, the predominant long edges in these images correspond to the point in the difference images between the positive and negative ridges, i.e. the center of the object. When the positive and negative are seen in the difference image, a long edge is also seen located at the outer boundaries of these areas, cf. Figure 5.20a. When the definition of the difference image is poor, as in Figure 5.19d, the resulting edges are also confused and directly yield little information regarding the leading and trailing edges, see Figure 5.20d. Figures 5.20f and 5.20g contain clear, long, connected edges which do correspond to the trailing edge and center location of the stimulus. These connected edges can be extracted as boundaries. The boundaries are indicative of the motion of the object between the images. Tracking the boundaries over time provides information regarding how the motion changes over time.

5.3.2.6 E9: Feature Matching - Moravec Interest Operator

The final experiment in the extracted-feature category illustrates the results obtained by applying a machine vision operator directly to tactile images. In this case, the interest operator developed by Moravec [137], [138] was used to locate points of interest in image n. The neighborhood around each of these points was correlated with image n+1 and the location of the best fitting neighborhood saved. The location in image n of the interest point and the location in image n+1 of the chosen match produce a vector describing the movement from one frame to the next of that point. The weighted average of all of the movement vectors is the estimate of the image motion between frames.

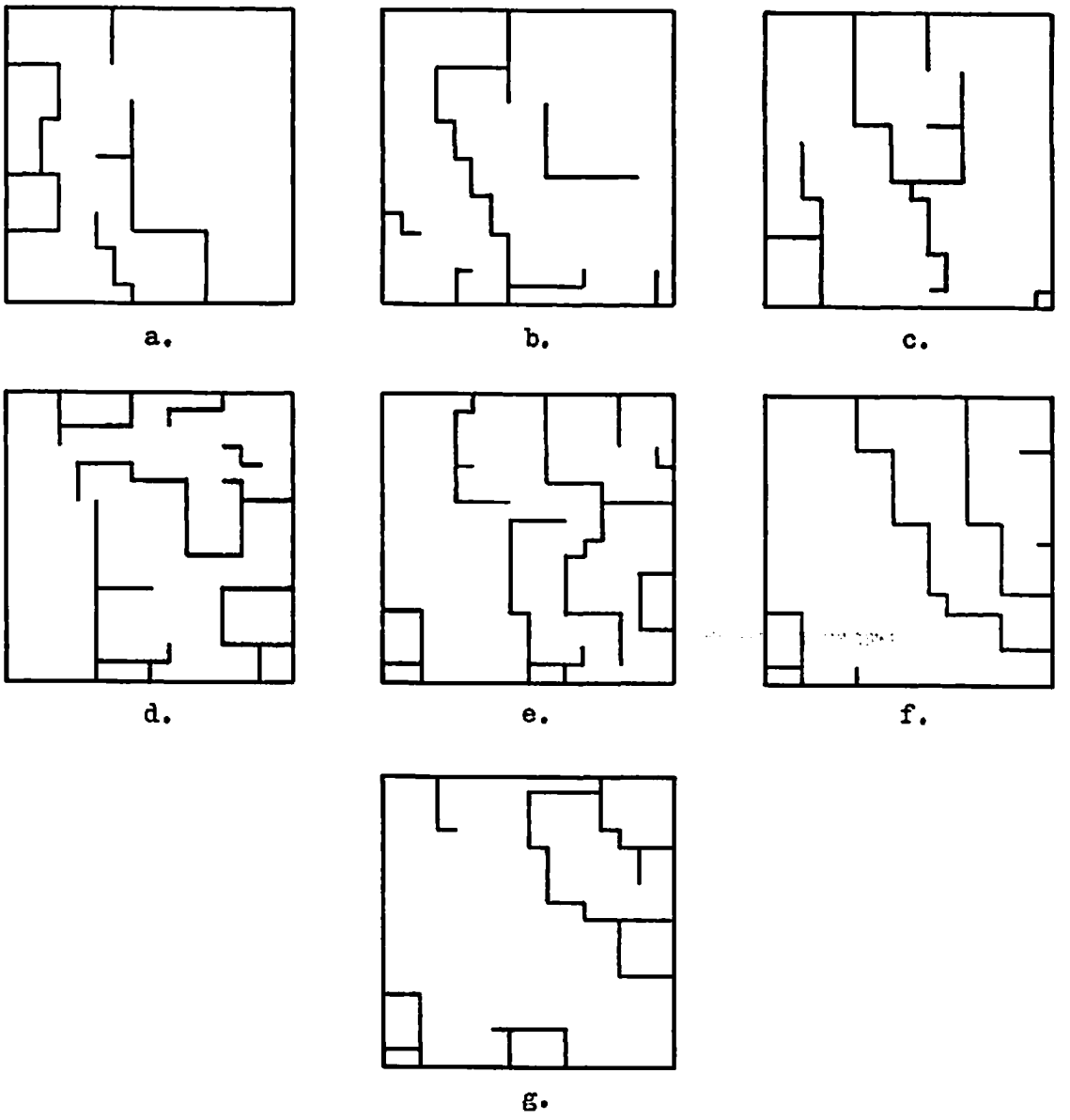


Figure 5.20

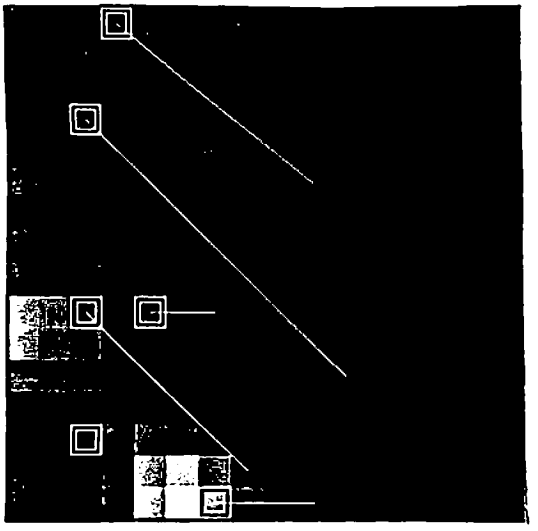
This shows the results of applying an edge relaxation on difference images shown in Figure 5.19. The process was allowed to iterate 10 times, a point at which very little change was observed in the probabilities.

The purpose of the Moravec operator is to select a set of locations uniformly distributed over the image where each location exhibits high contrast over its local neighborhood. The interest factor for a given point is calculated as the minimum of the sums of squared differences along the diagonal, horizontal, and vertical paths through the forcel in question. A 3x3 neighborhood was used for this experiment and a location is chosen as a feature if its interest factor is the maximum of the factors in its local neighborhood.

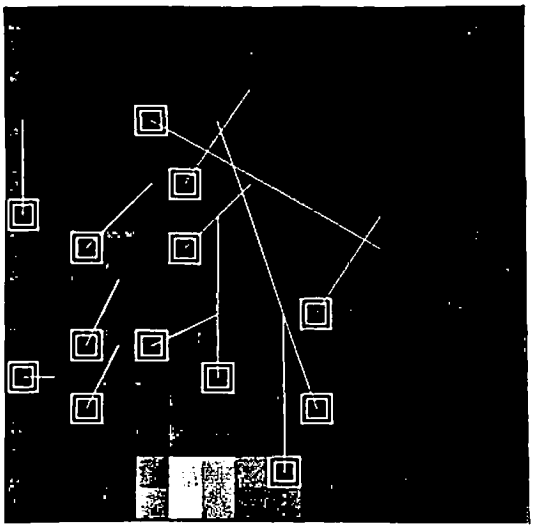
Cross-correlation of the window containing a feature point is used to locate the feature in the subsequent image. The maximum correlation value is also retained to be used as a weight in the average movement calculation. The final estimate of the motion is determined as the weighted average of the motions of all of the feature points whose correlation value is above 0.75. This threshold was chosen simply to select only strong matches. An automatic technique for choosing the threshold could be developed based upon, say, the distribution of the correlation values.

Figure 5.21 contains the selected features and their offset vectors for the eight cylinder images. In this case, a point's interest factor needed to be the maximum of those in a 3x3 neighborhood around the forcel in order for the point to be chosen. The cross-correlation used a 5x5 window. In the Figures, the intensity values composing image n are displayed with outlines indicating the forcels chosen as feature points by the interest operator. The vectors indicate the estimated movement of the features from one image to the next. The Moravec operator has the property of uniformly distributing the selected features over the image. This occurs as long as the image is not composed of a smooth, linear gradient. Since the force images do contain smooth areas, location in these areas are chosen as features, cf. Figure 5.21e and f. From the characteristics of the tactile domain it is known that these points are not on an object thus their motion estimates will produce confusing information and should not be considered. Also note the large displacements apparent for some of the features in almost all of the images: these are clearly ridiculous estimates of the motion. Again from the characteristics of the domain, we know that the motion in the sensor images is equivalent to the object motion. Since one can assume knowledge of the motion of the sensor, limits on the motion of objects can be determined. Thus the area of search in image n+1 for location of a feature can be limited. Moravec used the same approach in the visual domain. That is, an upper bound can be placed on the inter-frame motion, thus allowing search to be limited. This is explored below. The results of the experiment are too poor to be of any practical use; however, modification of some of the parameters produces improved results.

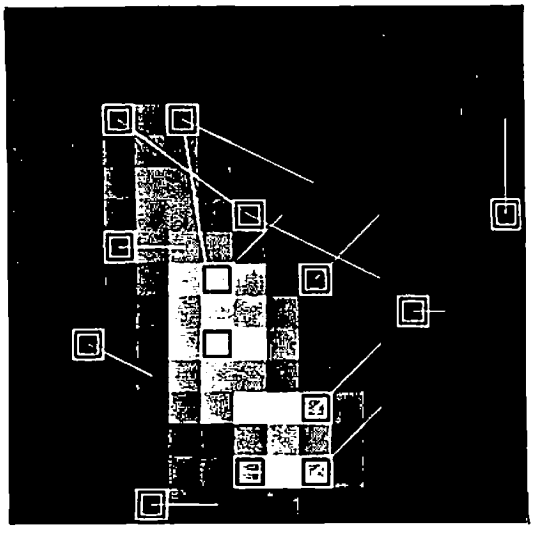
Figure 5.22 contains the results of the same algorithm applied with one parameter changed. In this case a forcel is chosen as an interest point only if its interest factor is the maximum of a 5x5 neighborhood centered at that forcel. Again a 5x5 window was used by the correlation procedure.



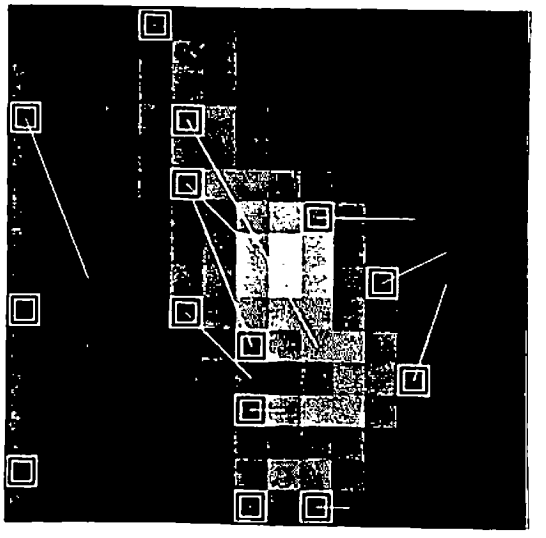
a.



b.



c.



d.

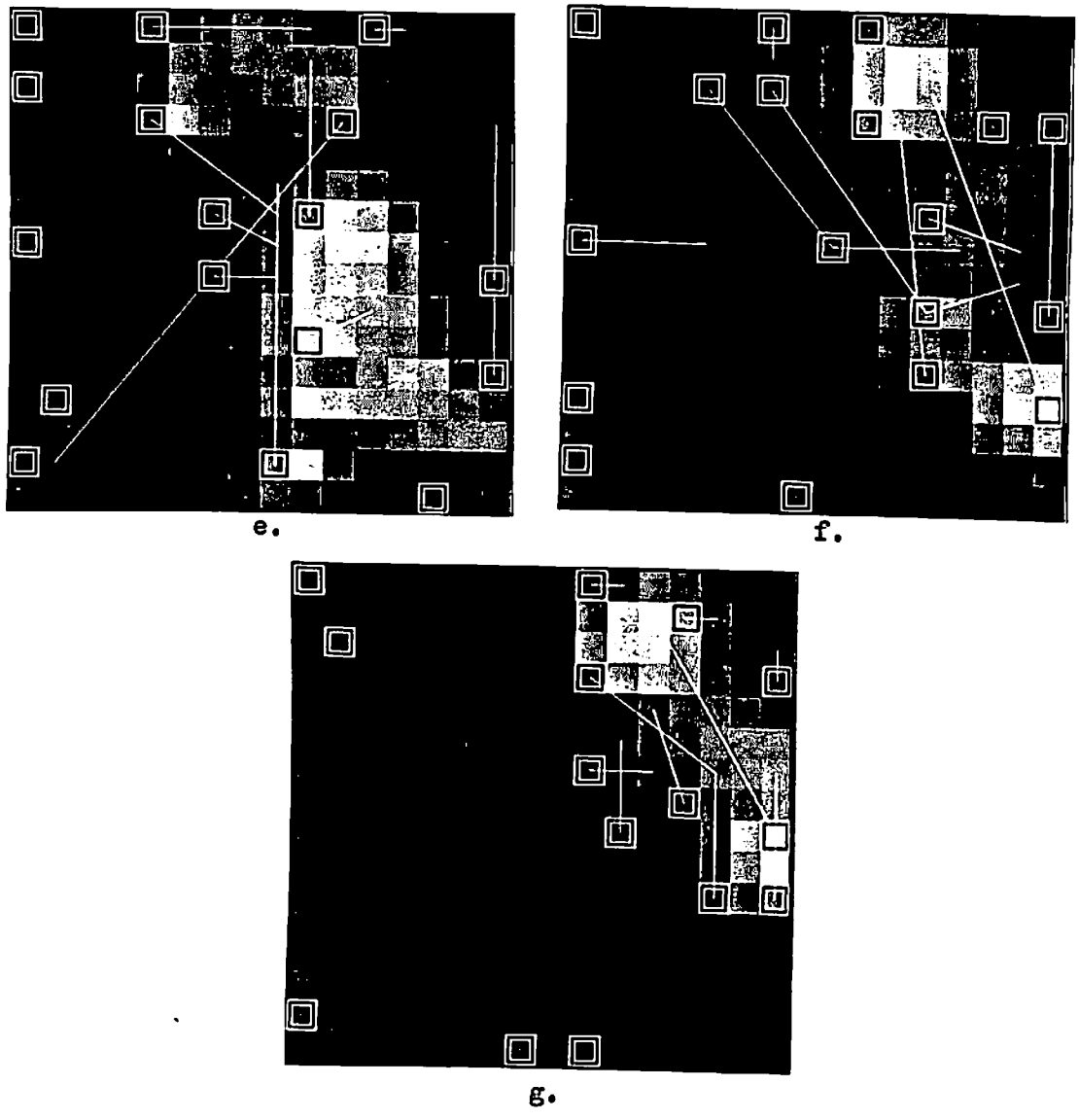
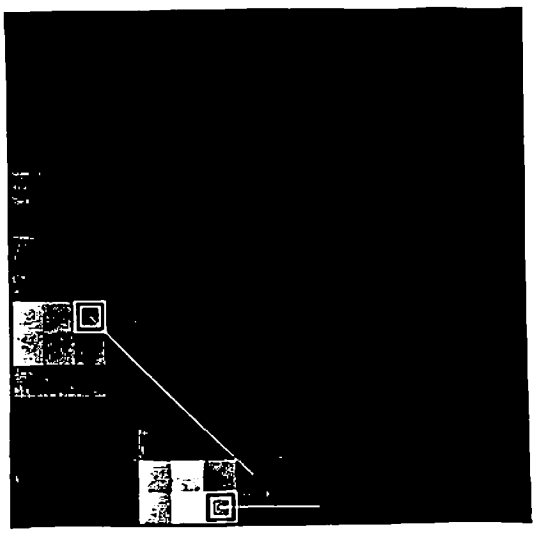
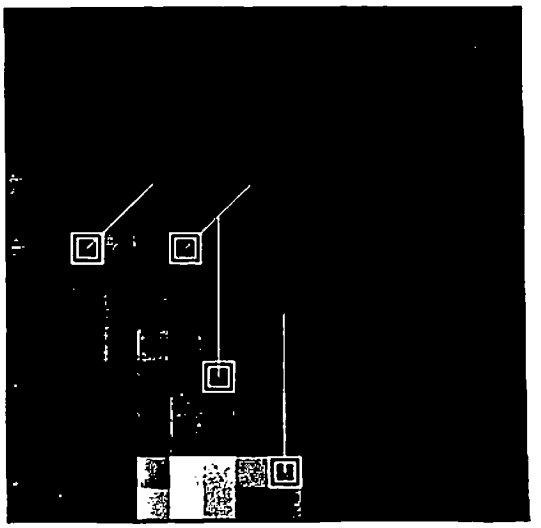


Figure 5.21

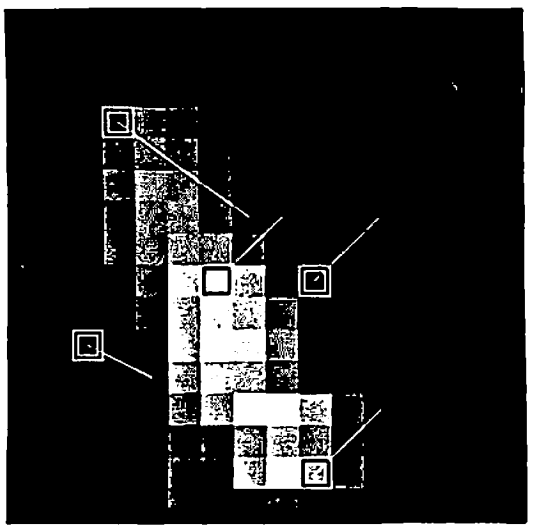
These images are the original tactile images shown in Figure 5.2 with boxes demarcating the forcels chosen as features by the Moravec operator. A 3x3 window was used during the feature selection process. These features are correlated with the next image and the location with the highest correlation value used as the estimate of the new location. The vectors connect each feature with the location in the next image which produced the best match.



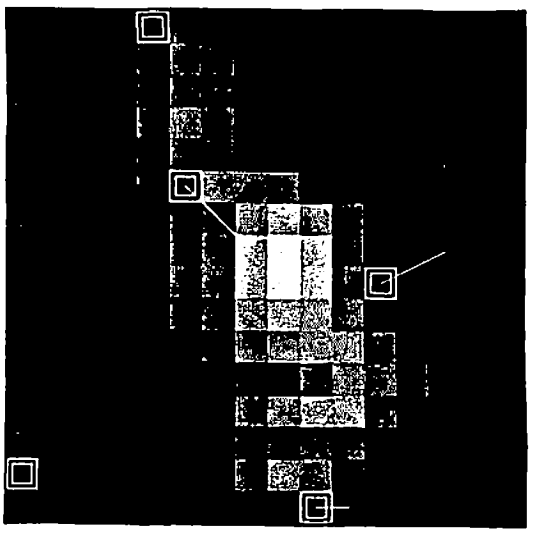
a.



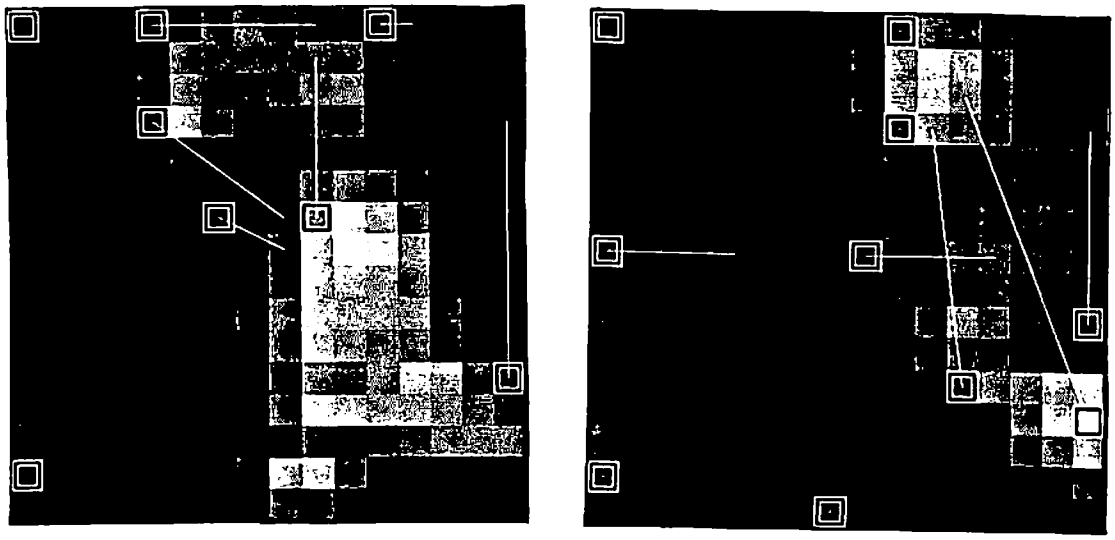
b.



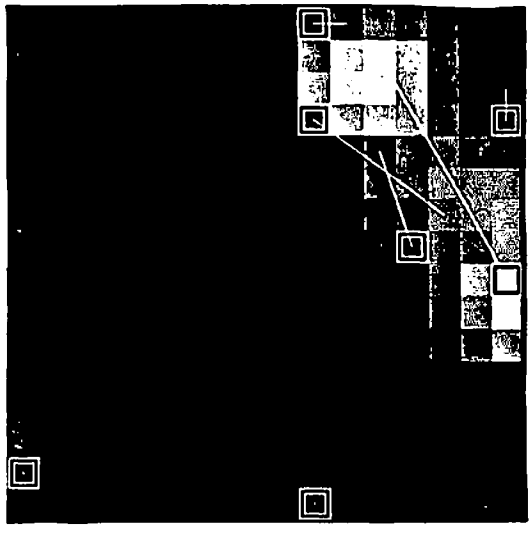
c.



d.



e. f.



g.

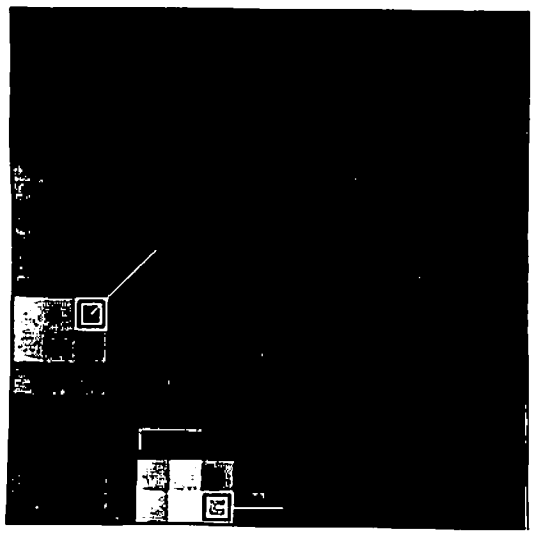
Figure 5.22

This figure contains the results of the feature-matching process applied with only one parameter changed. As in Figure 5.21, the Moravec operator was used to select features. Here, a location is chosen as an interest point only if its interest factor is the maximum of a 5x5 neighborhood centered at that forcel.

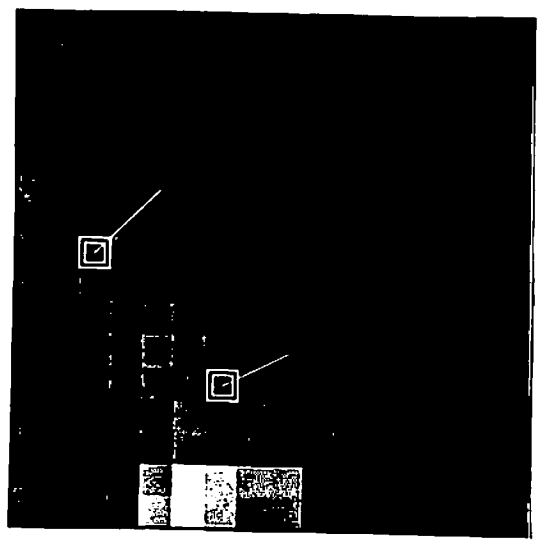
The effect of increasing the neighborhood size for the feature point selection is to reduce the number of points chosen as interest points. While this improves the results to a degree, the problems mentioned above remain. These include the fact that points not on the object are chosen as interest points and these points are poor motion indicators since they correlate with noise in the background. Further, the location of some feature points in the second image result in unrealistically long motion vectors.

The final set of images, Figure 5.23, contains the chosen features and vectors where the feature selection has been limited to points on the object and the search in image n+1 for a given feature has been limited. The objects are defined as the areas above threshold where the thresholds are those chosen in E4. The feature selection phase uses a 3x3 neighborhood and the cross-correlation phase again uses a 5x5 window. The upper limit on the size of the possible motion vector components is two forcels, that is, the search area in the second image was a 5x5 neighborhood.

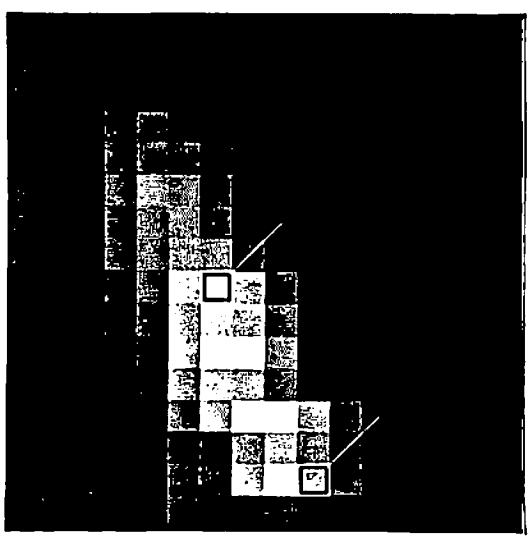
Notice that the selected features all reside on the object as defined by the binary images in Figure 5.14 and that they are not exactly the same features as chosen in the previous versions of this experiment. Note also that the motion vector more accurately reflects the motion of the object when the search neighborhood is restricted. With the exception of a few features which correlate best with themselves in the next image, i.e. a zero offset vector, the motions are reasonably accurate. Such features are found at the edges of the tactile images and correspond to features which actually move off the sensor. The overall average motion vector is (0.09,0.05) which is a reasonably accurate estimate compared with the actual vector of (0.1,0.05).



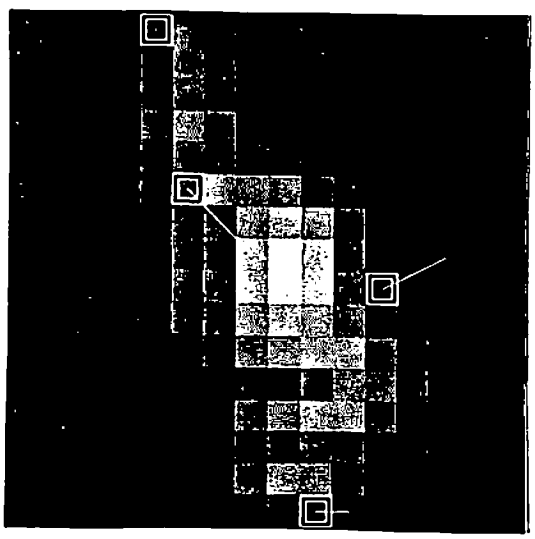
a.



b.



c.



d.

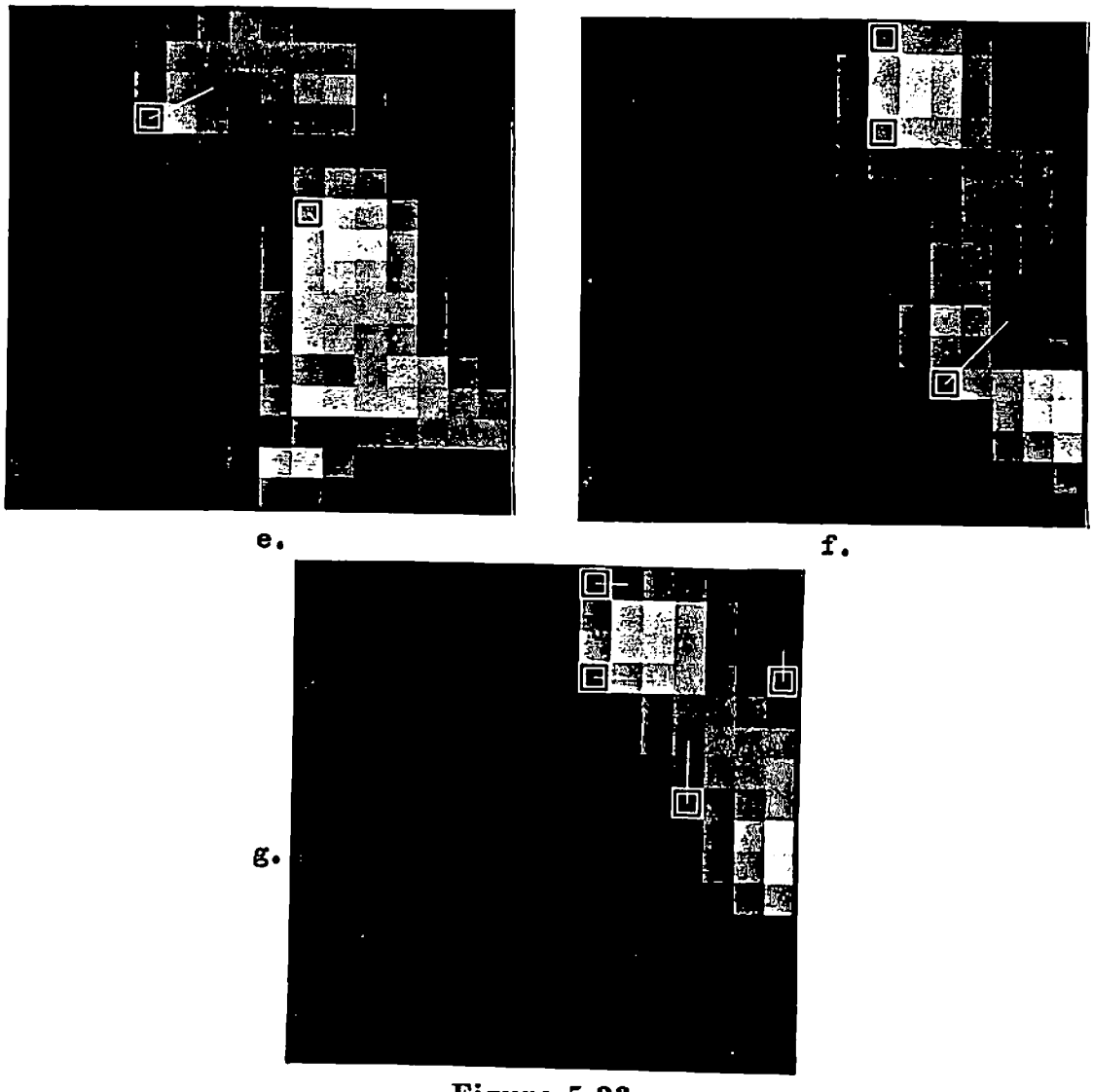


Figure 5.23

This set of images contains the chosen features and vectors where the feature selection has been limited to points on the object and the search in image n+1 for a given feature has been limited. The objects are defined as the areas above threshold where the thresholds are those chosen in E4. A 3x3 window was used during the feature selection phase and a 5x5 window during the cross-correlation phase. The upper limit on the size of the possible motion vector components is two forcels. That is, the search area in the second image was a 5x5 neighborhood.

5.3.3 Surface Model Construction

The experiments discussed in section 5.3.2 and 5.3.3 were intended to investigate the extraction of motion information from a sequence of tactile images. Let us now turn attention to the use of multiple images for the construction of a more accurate and complete model of an object's surface than is available from a single image. Consider the tactile images in Figure 5.2a and h and their respective schematic representations in Figure 5.1a and h. In both cases the stimulus object is only partially in contact with the sensor. The tactile image contains only a section of the possible complete imprint. In this case the object being sensed is smaller than the sensor surface; thus, it is possible to position the sensor so as to receive a complete imprint, cf. Figure 5.2c,d. In general, the object being sensed is larger than the sensor. Here, multiple images must be merged in some fashion to produce a coherent model of the object's surface. This process requires two steps. The first consists of registering the data images spatially. The second includes the actual modeling of the object's surface from the sensor data. The following experiment contains two versions of this theme and results in a rough model of the surface of the cylinder. The model produced consists of a surface patch which describes the data as opposed to an abstract model.

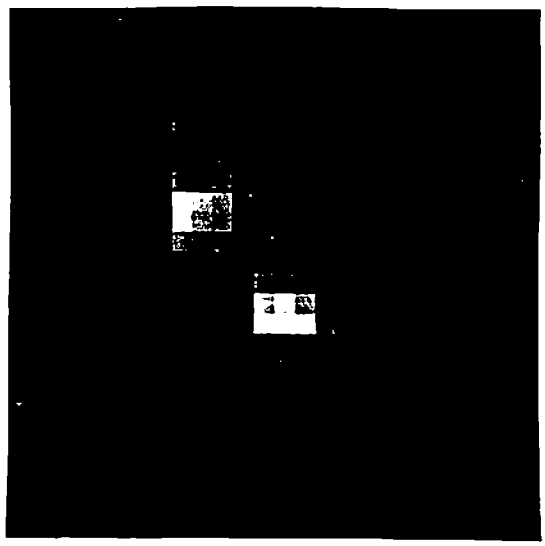
5.3.3.1 E10: Surface Model from Multiple Tactile Images

This experiment builds a model of the surface of an object from the moving cylinder sequence shown in Figures 5.1 and 5.2. The object in this case is smaller than the sensor surface, but the technique described here applies to objects which are large relative to the sensor surface as long as the inter-image motion is kept small relative to the size of the sensor.

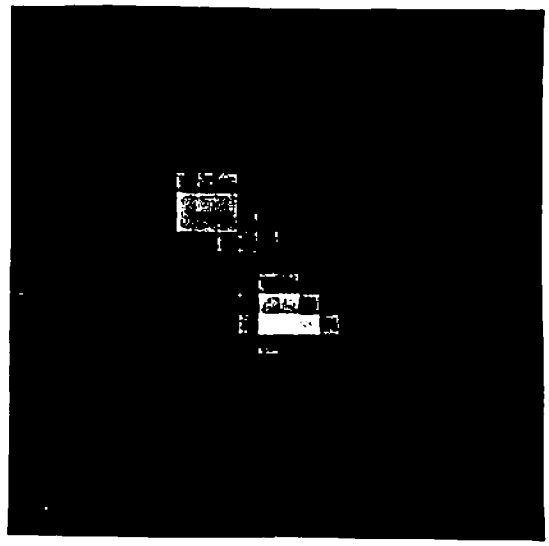
The registration between frames consists of finding the offset between images n and n+1 for all successive pairs in the sequence. The offset is then used to align the images which are combined to form a new image. Experiment E1 above consisted of finding the offset vector which resulted in the highest cross-correlation value between a pair of images. These offset values are exactly the quantities needed here. The cross-correlation technique of that first experiment is, in fact, a registration technique.

The offset vector calculated for the first pair of images is used to align these images and a new image is produced by averaging across both images for every forcel shared. That is, the averaging occurs across the two images in the area of overlap. This process repeats for the successive images with the offset always applied to the previous image coordinate system and the averaging for a given forcel including contributions from only those images which overlap at that forcel. Figure 5.24 contains the intermediate and final results of this processing.

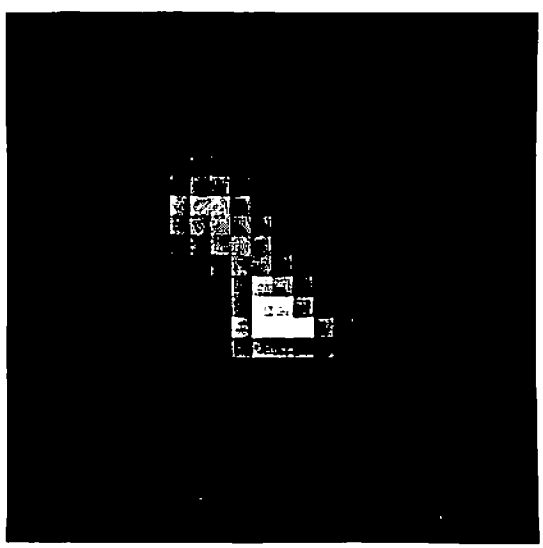
Figure 5.24a contains the first image in the sequence positioned on a dark field so that the other images can be combined in an obvious graphical representation. Each of the subsequent images contains the previous combined image and the next image from the sequence with Figure 5.24h containing all



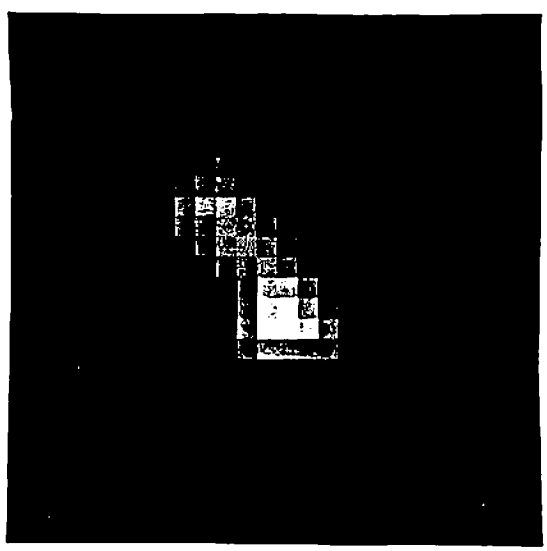
a.



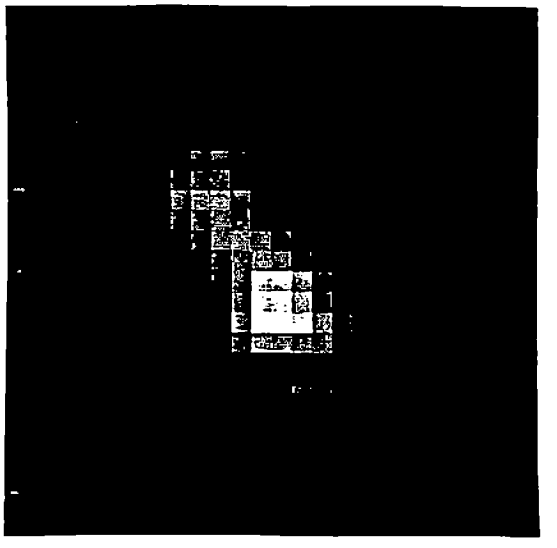
b.



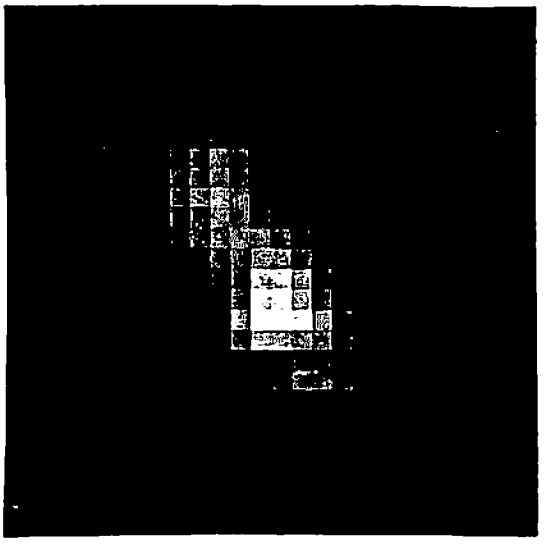
c.



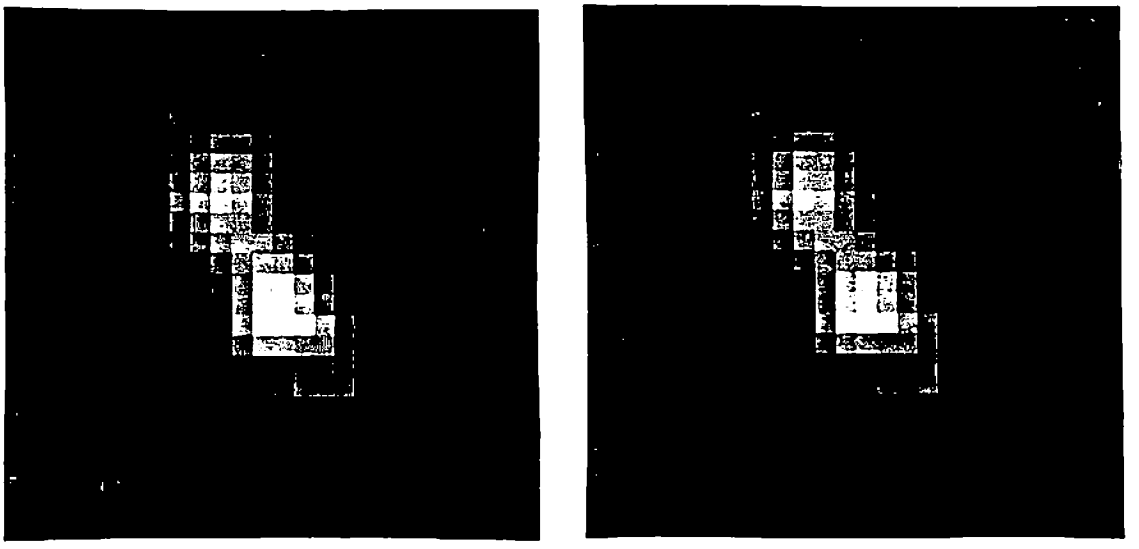
d.



e.



f.



g. h.

Figure 5.24

This figure contains the images in the sequence positioned on a dark field so that they can be combined in a graphical representation. 5.24a: This is the first image from the sequence. 5.24b: Here the second image in the sequence has been superimposed on the first with an offset calculated by the cross correlation technique of E1. 5.24c: This shows the first three images. The values in the combined image are produced by averaging the values of all images contributing to that point in the combined image. 5.24d: Images 1 through 4. 5.24e: Image 5 added. 5.24f: Images 1 through 6. 5.24g: All but the last image. 5.24h: All images combined.

eight images. Notice that the final image contains a long thin ridge as expected from the stimulus. Also notice that this image appears wider, longer, and to have less clearly defined edges than one would expect from a cylinder of the size used in the experiment. This is due in part to the discrete nature of the offset vectors produced by experiment E1.

Figure 5.25 contains the final image produced by the combination phase, Figure 5.24h, with each of the original images outlined to show its relative location. The horizontal shifting of the fourth, fifth, and sixth images illustrates the errors produced in part by the quantizing of the offset vectors.

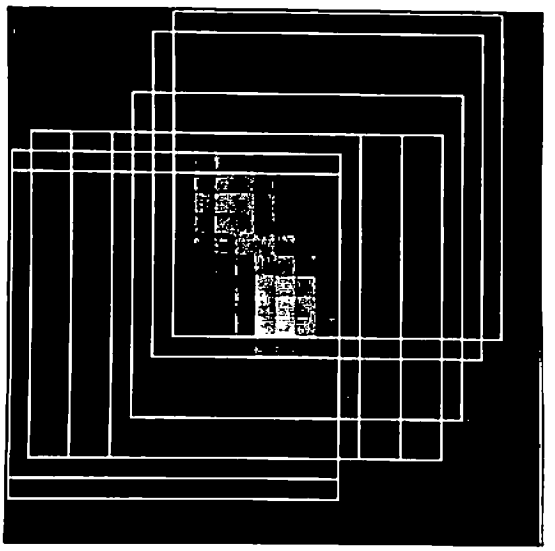
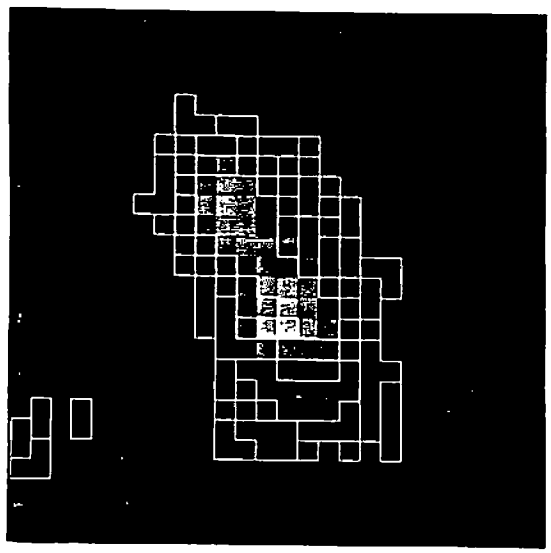


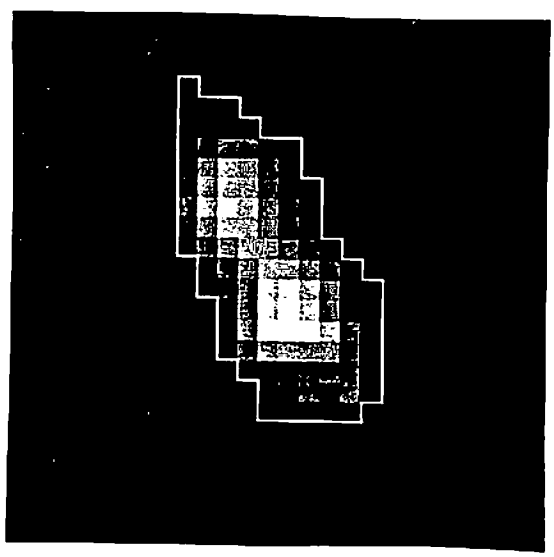
Figure 5.25

This shows the final image from the combination phase with the outlines of the original images included. The offsets for the images were determined by the cross-correlation process of E1. Notice that the single region in the combined image does not appear in full in the images at the beginning and end of the sequence.

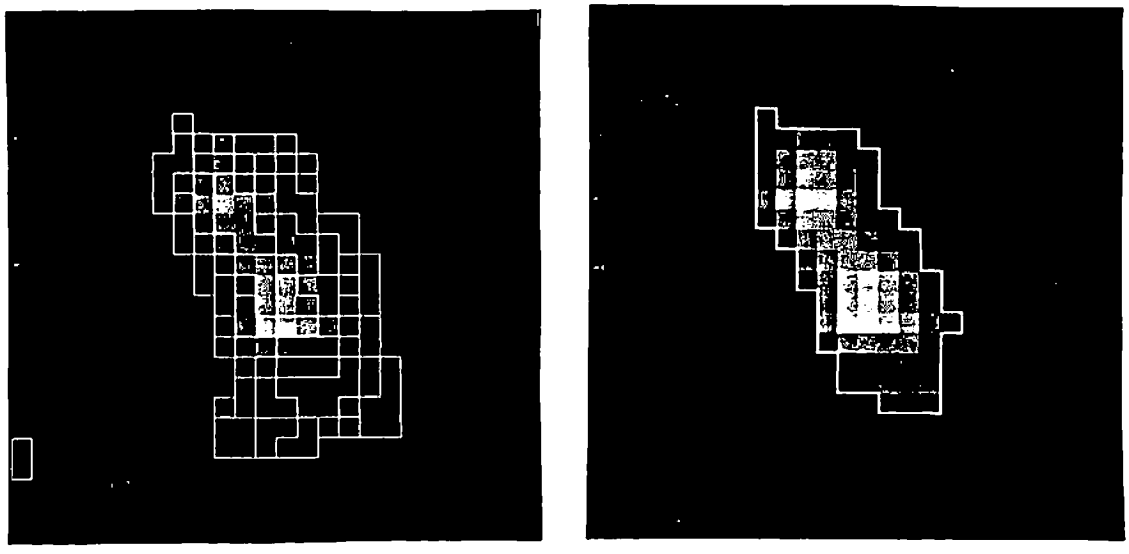
The results of this experiment can be used to illustrate an interesting point. Recall that E4 consisted of thresholding each of the images to pick out what areas in the image corresponded to the object. Viewed slightly differently, the area above threshold was labeled as an object imprint and had associated with it some confidence that it was, in fact, the result of contact with an object. Those threshold regions can be superimposed over the final image in Figure 5.26. For a given forcel, the confidence that that forcel represents a part of the cylinder's imprint is supported by the number of images in which that forcel is above threshold. Figure 5.26 contains the graphical results of this process.



a.



b.



c. d.

Figure 5.26

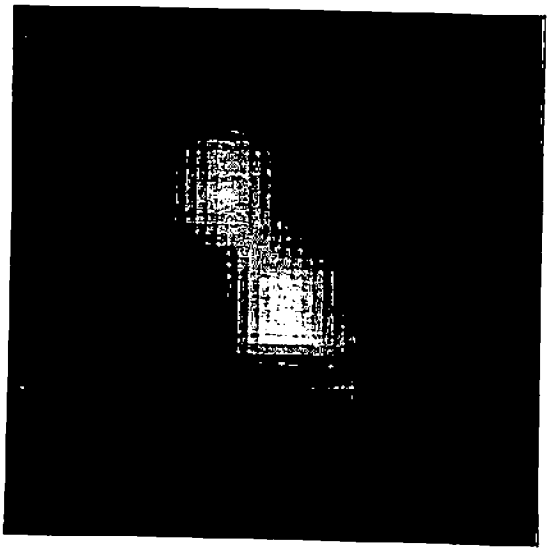
This shows the final combined image from Figure 5.24 with the object separated from the background. In 5.26a, a threshold value of 25% of full scale was used to separate the object from the background in each of the original images. A forcel is outlined if it was above threshold in any of the original images contributing to that forcel in the combined image. The forcels along the ridge are above the threshold in all eight original images. Figure 5.26b contains the final combined image with the region above the 25% of full scale threshold outlined. Figure 5.26c and d contain similar results except that the threshold used was 35% of full scale.

In Figure 5.26a, a threshold value of 25% of full scale was used to separate the object from the background. The forcels along the ridge are above the threshold in all eight original images. There is, therefore, very high confidence that these forcels are part of the image of the cylinder. The forcels farther from the ridge are above threshold in fewer and fewer of the data images and thus the confidence that they are a part of the cylinder's image is lower than for those along the ridge. Figure 5.26b contains the final image with the region above the 25% of full scale threshold outlined. Figure 5.26c and d contain similar results except that the threshold used was 35% of full scale. As one would expect, the above threshold region reduces in size as the threshold increases. A more intelligent technique would include a weight or confidence factor for each forcel based upon the number of original images in which the corresponding forcel was above threshold.

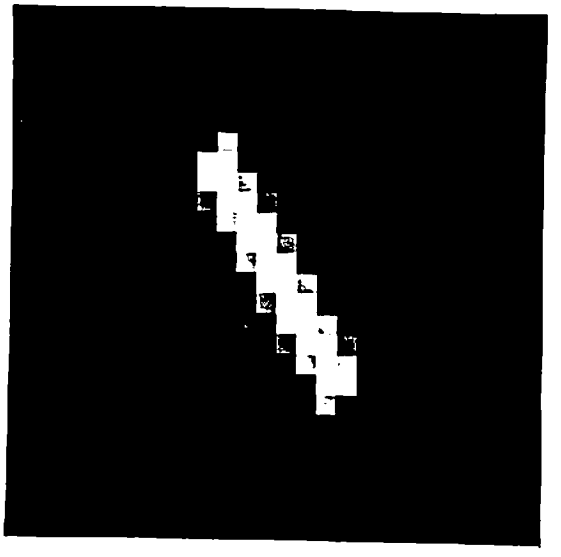
The second phase of the multiple image modeling procedure requires that some form of object model be produced from the merged data. A number of modeling techniques have been proposed for use with graphical modeling systems, CAD systems, and machine vision systems (see [23] for a review of approaches to the design of model-based vision systems.) The underlying primitives have included planar facets, generalized cylinders, cones, swept bodies, [35] surface patches [221], [222], [223], etc. The approach shown here uses cubic b-spline surface patches [51] with the data points in the merged image, cf. Figure 5.24h, providing the control points for the patch. Figure 5.27a contains the b-spline surface patch of degree 3 generated from the data. The surface patch has the property of smoothing extremely sharp discontinuities in the data yet retaining a great deal of local control over the shape.

Figure 5.27b contains exact data from a modeled cylinder at the same angle of inclination as the stimulus. The diameter and length have been increased by one forcel to make the comparison with Figure 5.27a more realistic. 5.27c contains the b-spline surface patch constructed using the exact data as control points. One point to notice here is the degree to which the sensor and processing algorithms tend to widen the image. The attempt focused not on matching the calculated surface patch to an idealized model but rather on constructing a model from experience. It is this "experience-generated" model which is matched against new sensations. The exact data are included here for comparison purposes only.

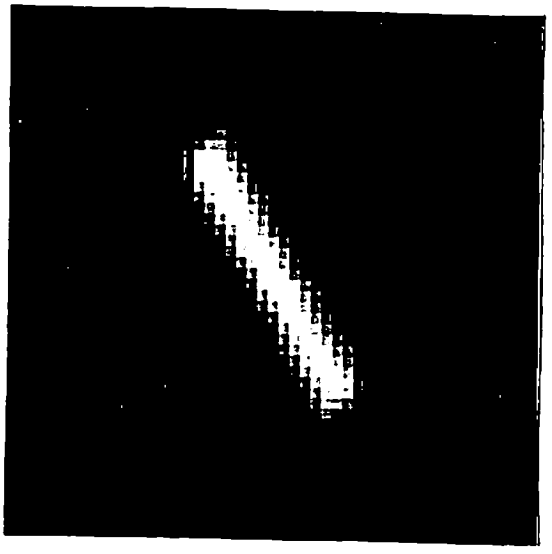
One of the problems with the previous experiment was the coarseness of the offset vectors as calculated by the cross-correlation method of experiment E1. An obvious extension to this experiment consists of using the offset vectors produced from the cross-correlation technique applied to the interpolated data, cf. experiment E3. Recall that the data images were used to produce images in which each original forcel was replaced by a 3x3 window of values linearly interpolated from the 3x3 neighborhood around the original forcel. The offset vectors resulting from the cross-correlation technique were slightly better estimates of the motion than those from experiment E1.



a.



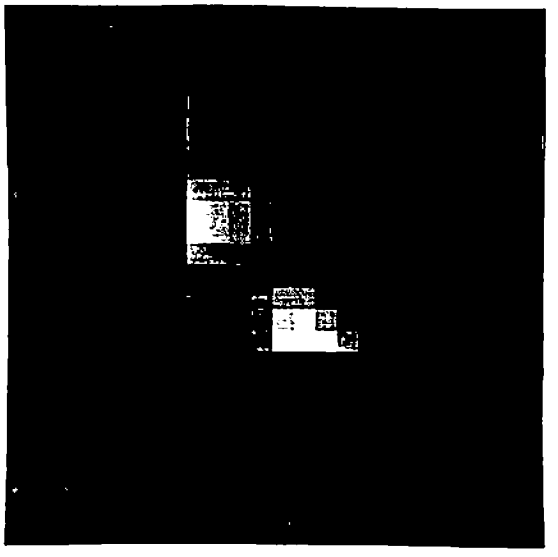
b.



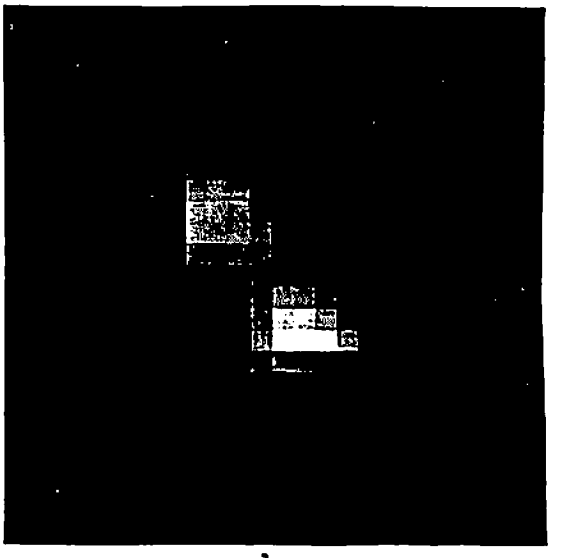
c.

Figure 5.27

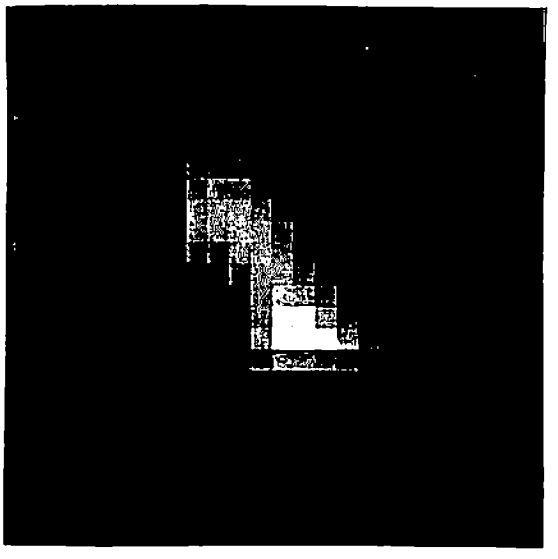
This figure shows a b-spline interpolation of the merged data as well as perfect data for a stimulus and its surface patch interpolation. 5.27a contains the b-spline surface patch of degree 3 generated from the data produced by merging the 8 cylinder images, cf. Figure 5.24h. 5.27b contains exact data from a modeled cylinder at the same angle of inclination as the stimulus. The diameter and length have been increased by one forcel to make the comparison with Figure 5.26h more realistic since the offset vectors were quantized. 5.27c contains the b-spline surface patch constructed using the exact data as control points.



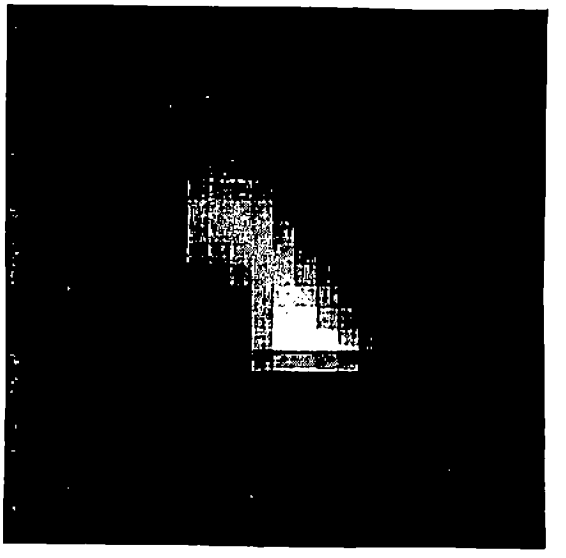
a.



b.



c.



d.

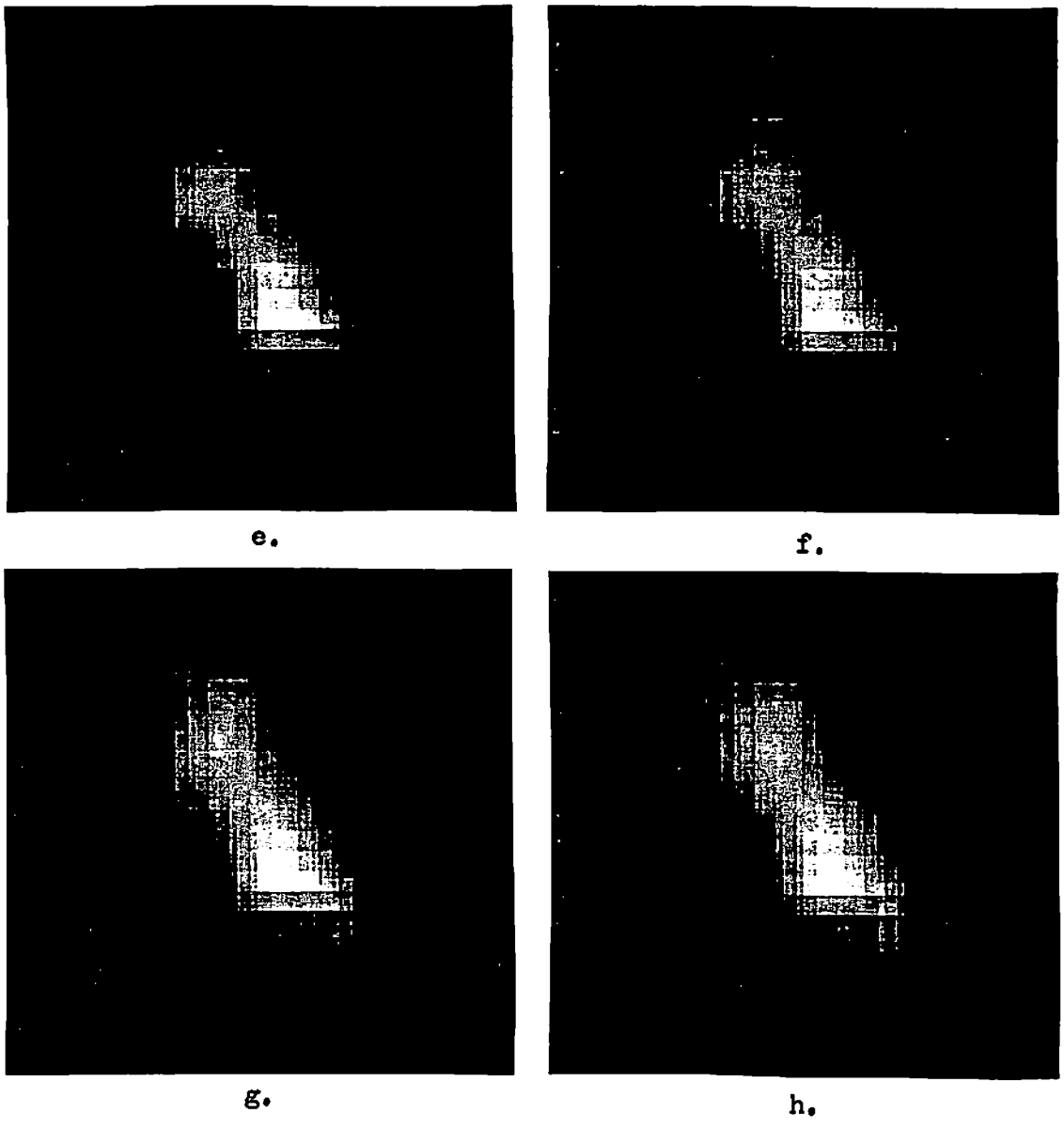


Figure 5.28

This figure contains the results of the merging process shown in Figure 5.24 applied to interpolated data. 5.28a shows the interpolated image resulting from the first cylinder image, cf. Figure 5.2a. The remaining images show the superposition of the seven other interpolated images using the offset vectors from experiment E3.

Figure 5.28 contains the results of the merging process and are analogous to those in Figure 5.24. Figure 5.28a shows the interpolated image resulting from the first cylinder image, cf. Figure 5.2a. The remaining images show the superposition of the seven other interpolated images using the offset vectors from experiment E3.

The final image, Figure 5.28h, exhibits the same structure as is apparent in Figure 5.24h with slightly less detail since the interpolation process has the property of blurring the data. Figure 5.29 contains the merged image with the outlines of the images superimposed. Notice that the quantization problems in the offset vectors still exist.

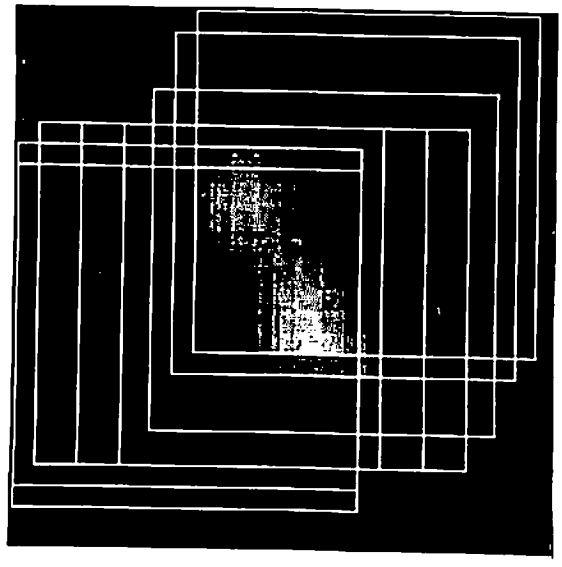


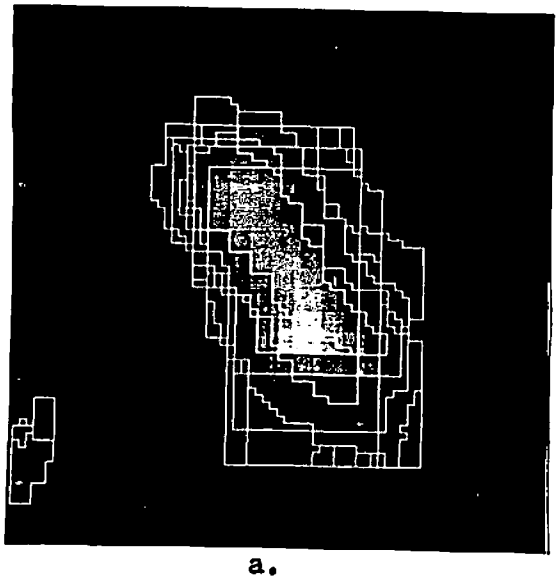
Figure 5.29

This shows the final image from the combination phase applied to the interpolated data with the outlines of the original images included. The offsets for the images were determined by the cross-correlation process of E3.

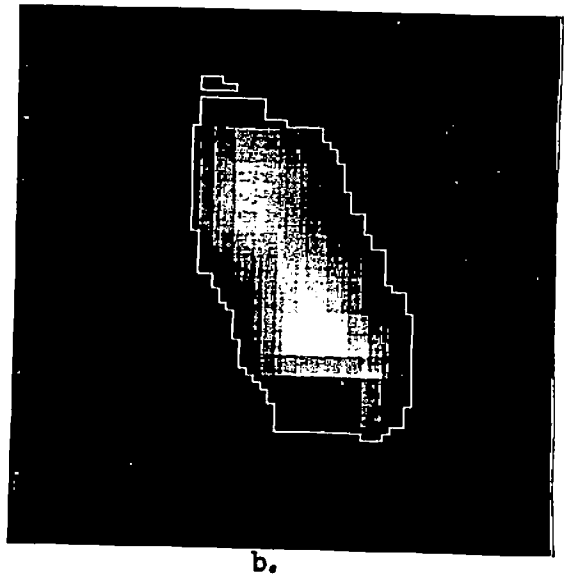
Figure 5.30a illustrates the boundaries of the regions which are above 25% of full scale. The extent of the regions is essentially identical with those from the uninterpolated data images with the only exceptions appearing as fine subdivisions along the edges. Figure 5.30b exhibits smoother boundaries than Figure 5.26b yet the area is larger. This is due to the blurring effect of the interpolation process. Figure 5.30c and d are the result of the same process using a threshold of 35%. The boundaries are smoother than those in Figure 5.30d and the areas are roughly equivalent. Figure 5.30e contains the outline of the region in the merged image which is above the 50% of full scale value.

Figure 5.30

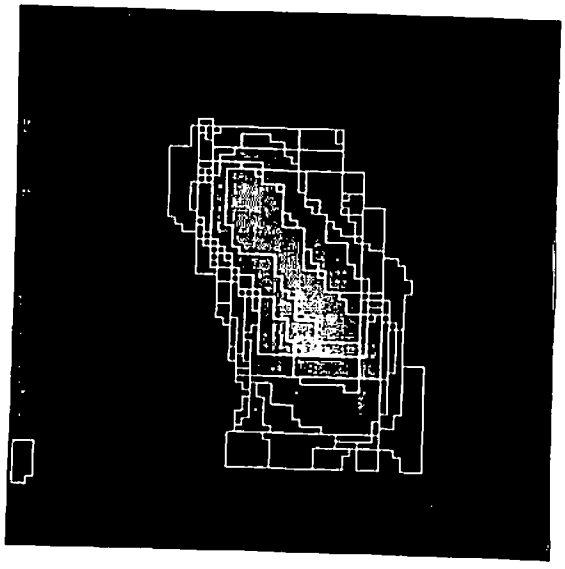
This figure illustrates the boundaries of the regions in the original interpolated images which are above 25% of full scale. 5.30b shows the boundary of the region in the combined image which is above 25% of full scale. Figure 5.30c and d are the result of the same process using a threshold of 35%. Figure 5.30e contains the outline of the region in the merged image which is above the 50% of full scale value.



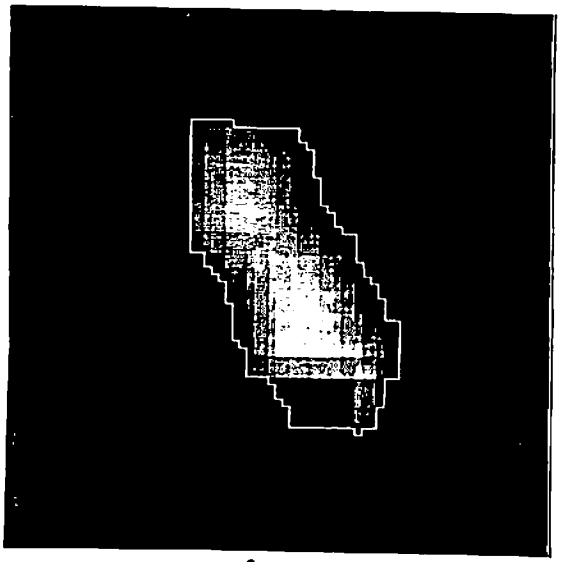
a.



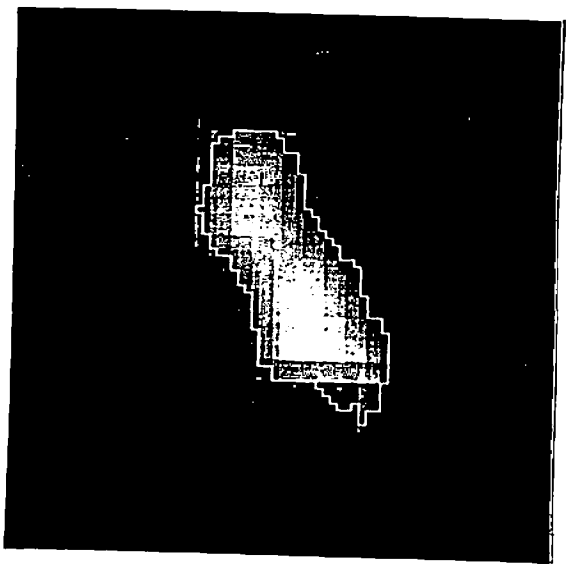
b.



c.



d.



e.

This region represents those forcels with high confidence of inclusion in the image of the cylinder and has an elongated shape with relatively smooth boundaries. The results of experiment E3 were not significantly better than those of experiment E1 and thus it should not be expected that the results obtained here with the interpolated images should be much better than those obtained with the original images. The improvement that was seen came in the form of slightly smoother boundaries.

This experiment has shown the results of one technique for merging multiple tactile images and then modeling the surface in the resulting image. This technique also applies to objects which are larger than the sensor.

This chapter has explored some of the issues surrounding the area of dynamic tactile sensing. Dynamic tactile sensing consists of the acquisition and processing of sequences of tactile images collected from the same sensor and separated temporally by small increments. The characteristics of the domain were discussed, as were various approaches to processing of tactile image sequences. Algorithms using both correlation and feature-driven approaches were presented. The processing can be used to extract information regarding the motion of objects contacting the sensor and to build more complete models of the surface of an object being sensed. The information extracted from the sequences of images must be coupled with the information from other sensory systems and movement control by a high-level control and planning mechanism [10]. Such a mechanism is the topic of the next chapter.

Chapter 6

A Schema Approach to Robot Control

6.1 Introduction

The previous chapters have discussed senses, tactile array sensor design, and algorithms for static and dynamic tactile image processing. The result of this work is a sensor and some insight into the extraction of useful information from the tactile array data. The development of robotic tactile sensing requires not only an understanding of the tactile domain but also the integration of the information gained from touch and vision into the control structures of the robot. The areas of knowledge representation, high-level planning, control, and sensory processing are all elements of this process. A framework is developed here which includes elements from all these areas and tightly couples action and sensory perception.

Research on knowledge representation has yielded a number of generic approaches including the use of the predicate calculus as a representational notation, production systems, frames, schemas, and semantic networks (see e.g. [28] for an overview). Most of these approaches and systems function with two basic assumptions: complete knowledge of the domain and a static world. The language systems tend to work in domains wherein the knowledge is contained either explicitly in the form of information about elements and constructs of the language or implicitly in the form of inference mechanisms. Thus decisions about the language can either be made from the representation itself or it is known that they cannot be made.

A robot functions in a real environment, and no matter how simple the domain, complete knowledge about the world is unavailable. Robot control systems must function with information derived from an "internal model" which provides a partial representation of the world and sensory information derived from the investigation of the environment. It is inconvenient, or even impossible, for the model to contain detail at the level existing in the actual world. Further, the sensory information available to the system is also incomplete since only a portion of the environment can be scanned and the data are of finite resolution and include inaccuracies.

The second underlying principle on which many knowledge representation systems rest deals with the temporal nature of the world. Most systems consider the world to be inherently static. That is, they deal with static relationships between entities. Physical objects in world models or words in linguistic models typically comprise the elemental units in the representation with spatial relationships comprising the links, or connections, between the elements. These "spatial relationships" can be in the form of actual physical relationships in models of the environment or grammatical relationships in language systems. All of these relationships are viewed as being static in nature with change represented as operators which function on one representation to transform it into another.

The world of the robot, however, is rarely static. Sensory information changes continuously as does the relationship between the robot system and the world. A robot "brain" must be capable of representing not only the dynamic nature of the robot's environment and the robot's interaction with the environment; it must incorporate the dynamic nature of the environment in order for the behavior produced to relate to the world in any meaningful way.

Distinctions are often made between "planning" and "control". Planning can be defined as "... a search for a series of actions to bring about a particular desired world-state", [28] and need not necessarily be a real-time process. A robot planner has the responsibility of determining the long-term behavior of the robot which will ultimately allow a particular goal to be satisfied. Control, by contrast, can be defined as the moment-by-moment supervision and modification of the processes within the robot system, servo control of the actuators being a very low-level example. Yet the above distinction between planning and control becomes blurred at some intermediate level where control becomes abstract enough to meld with the planning function. The approach developed here is intended to provide control at all levels extending from specifying the motion parameters of a given joint to long-term planning.

A robot system consists of an articulated mechanical device capable of interacting with the environment, sensors of various sorts, and a logical unit given the responsibility of controlling the system. In order for the systems to be useful, several characteristics are needed. Robots, lacking self-motivation, require direction from an external source, usually humans. If we assume that the user is to program the machine at an abstract level relative to the individual motors, then stable control of the actuators is needed and is assumed to exist in this chapter.

A robot "brain" needs a representation of the world (in some form) a mechanism for planning actions, and a system which provides stable control of the mechanical aspects of the robot. However, these alone are insufficient for a machine to interact with the world. The final component is the sensory processing. The robot must be capable of obtaining information about the current state of the world. This question is sometimes skirted by assuming that the world model is updated more or less continuously independently of the robot system. One of the goals here is to specify how sensory information can be coupled into control and planning at all levels of abstraction. Position and torque feedback may be required to provide stable control of a particular joint; tactile sensory information may be required to direct the final stages of a grasping action, while visual inspection of the environment may be needed for high-level planning.

The major goal is to offer a method for specifying the task to be performed and the salient features of the environment. A great deal of effort has been directed toward the development of robot programming languages, cf. [205], [139], [63], [126] and the review in [196]. Languages such as AML for the IBM robot system and VAL for the PUMA robots provide communication between the machine and the user and handle questions such as trajectory calculations.

They provide "motion primitives" with which the programmer builds applications programs. The structures given in this chapter are presented in a pseudo-AML language to make clear how the ideas extend current robot programming systems. However, the intent here is not to develop a programming language but to specify a framework for developing languages for sensory-motor integration.

Robot programming languages comprise one necessary component of the type of system of interest here. The overall system can be thought of as a development environment (DE) which aids a human user in the construction of sophisticated robot behaviors. The DE contains three abstract structures of which the user is cognizant: schemas, goal structures, and expected sensory input structures. The schemas provide the focus of this work with the other structures fulfilling subordinate roles. It will be seen that robot behavior results from the activation of schema assemblages.

The function of the DE is twofold: to assist the user in constructing schemas and to handle the "mechanical" details of schema implementation. During the construction of schemas, the DE prompts the user with a prototype schema for the parameter and structural changes needed to implement a particular schema. Once completed, the DE adds the schema to the total system by assigning processing and communication resources. Resource management, communication pathway assignment, and logistical matters are handled by the DE without intervention by the user. Thus the DE provides the user with a view into the schema-controlled robot world uncluttered by the implementation details associated with complex electro-mechanical systems.

This chapter focuses on the schema component of the DE since it is this component which affords the combination and utilization of sensory data in control and planning. A schema can be viewed as an abstract data type with sensory processing, action, and possibly learning elements.

Just as larger programs are built of smaller programs, so does this framework envision larger schemas built out of smaller schemas. At the highest levels, these schemas can be considered as fulfilling the planning function while at the lowest levels they provide servo control with sensory processing incorporated at all levels. They can be thought of as a form of knowledge representation in that the robot actions associated with a particular set of sensory input values are encoded in the schemas.

Sensory information is typically integrated at two very different levels by present robot controllers depending upon the type of information involved. At the servo control level, velocity and position information for a given actuator are used by the control algorithms in order to achieve stable performance. Force feedback information is used at a level involving the interactions of several joint/link pairs. The other extreme is exemplified by what might be called the "snapshot" paradigm for the use of visual information in which the actions of the robot are halted, processing of visual input is requested of another unit, this other unit collects a picture from a camera, processes the data, and returns a coded form of the information. The robot system then proceeds "blindly" based

on this new glimpse of the world. A major problem with this approach is the need for the environment to be relatively static and well specified. The functioning of an individual robot currently tends to be viewed as a serial progression of events. Only a single task or subtask can occur at any given instant which implies discrete switching to new subtasks and an inability to smoothly change attention from one set of activities to another.

Robots are machines which interact with the world. They require the abilities of mechanical articulation and sensing. Present control systems provide the lowest levels of motion primitives; however, high-level sensing and acting are typically viewed as separate functions. In addition, robots are capable of executing only a single task at a time thus imposing a serial structure on the control process.

All of these considerations enter into the design of a robot control system. In addition, there is a basic question regarding how knowledge is represented. At one end of the spectrum is the view that knowledge should be represented in a declarative manner. That is, knowledge about the world appears in the form of facts. The other end of the spectrum is typified by the view that all knowledge is procedural. In this case, knowledge is in terms of programs which define how a system uses or relates to the world in a given situation. The approach taken here may be termed procedural although certain information is included in a more or less declarative form. Another axis along which systems can be differentiated concerns the driving force of the system. Systems can be goal-, or concept-driven wherein the actions result from top-down processing beginning with some abstract set of goals. Alternately, systems can be data-driven, in which case they respond in a more or less reflex fashion to the environment. It is clear that the distinctions just mentioned are extremes along what are probably continuums. The systems which ultimately prove to be the most flexible and adaptive will, undoubtedly, possess characteristics which fall somewhere between the extremes. These are but two of a set of characteristics and questions surrounding the areas of knowledge representations, planning, and control. It is not the intent to review this work in detail here but rather to present a framework which addresses some of the issues which need to be considered during the development of a robot control system.

As humans, we are constantly sensing our environment. Our actions provide new sensory input which is used to modify our actions this, in turn, produces more sensory input. In addition to new sensory information, an action yields an expectation of certain types of sensory stimuli. This expectation, combined with the actual sensory input, provides a complex error signal used for verifying and fine-tuning our movements. By contrast, robot systems have, to date, lacked the sophistication to utilize sensory information in this ongoing manner. In addition, while the temporal nature of coordinated activity invites a serial analysis, both sensory processing and motor control involve much parallelism or concurrency in the activity of different subsystems. Moreover action and sensory processing components are intertwined: actions require direct sensory feedback and produce expectations regarding sensory input, while sensory

input provides the information which modulates and initiates further actions. This is the continual process known as the action-perception cycle [11], [142].

What is needed is a representation which encompasses this action-perception cycle by integrating the motor and sensory aspects of each action. The schemas developed here are intended for robot control and must incorporate the characteristics of their domain and include capabilities for specifying the robot's actions.

An interesting approach to the use of sensory information in the control of a robot has been developed by Albus and his coworkers at the NBS [8], [9], [18]. A particular task, either for a single robot or for an entire factory, is decomposed into a set of trajectories at varying levels of abstraction. At the lowest level the trajectories describe the desired motion of the actuators of the system while at the highest they may describe abstract phenomena, e.g. the economic state of a factory. The hierarchy is defined by a sequence of levels, each with processes which monitor certain error signals and produce control signal targeted toward specific lower-level processes. The schema approach offered here differs from this hierarchial approach in that while the schemas may form a hierarchial structure at certain times, global control is accomplished through the competition and cooperation of a varying number of schemas thus imposing no rigid structure on the schema assemblages.

6.2 Schemas

A schema is an abstract type which monitors certain aspects of the current situation and becomes active when the situation matches the expected state. The current situation is defined by sensory input, the ensemble of schema activity, and by the goals of the system. Once active, a schema can request actions and further sensing to be performed by the robot. The process of sensory input monitoring is carried on in parallel by any number of **perceptual schemas** with the activity of any particular schema determined by how well it fits the sensory input as well as by the influence felt from other schemas. The overall representation of a complex environment takes the form of a **schema-assemblage** rather than a single schema.

A schema is both a process and a representation. The internal representation of the relationship of the system to the world involves the parallel activity of all those schemas which receive appropriately patterned input. The resultant environmental representation interacts with those processes which represent the current task specification for the robot to generate the course of action which can provide the input to the various motor schemas directly controlling behavior. The schema-assemblage is continually updated as action progresses.

Sensory stimulation cannot guide movement directly. Rather, it must be interpreted in terms of objects or other "domains of interaction" in the environment [74], [75] (see also [64]). After [11], the term "perceptual schema" is used for the process whereby the system determines whether a given domain of interaction is present in the environment. Perception of an object (activating

perceptual schemas) provides access to motor schemas for interaction with the object, but does not necessarily involve their execution. Only a few aspects of the environment can at any time afford the primary locus of interaction. A process of planning is required to determine the plan of action, in the form of an appropriately coordinated control program of motor schema activation, on the basis of current goals and the environmental model; and the robot must continually make, execute, and update its plans as it moves.

A motor schema is a control system which continually monitors feedback from the system it controls to determine the appropriate pattern of action for achieving the motor schema's goals, (these will, in general, be subgoals within some higher-level coordinated control program). Since the controlled system may itself be variable, the motor schema must either be supplied with the relevant parameters of the controlled system (for example, a grasp schema needs to know the position, size, and orientation of what is to be grasped) or must itself estimate them using a learning procedure or "identification algorithm". The identification procedure may be viewed as a perceptual schema embedded within a motor schema.

Schemas vary in abstraction from those which function as the lowest-level servo-control systems for the individual actuators to those which control high-level functions such as assembly. In the pseudo-AML presentation of schemas, they may appear as single AML commands, as normal subprograms, or even as complex structures. The overall structure of the system at any instant consists of the coactivation of a number of competing and cooperating schemas. A schema, then, is an abstract representation of a collection of actions and sensory processing units as integrated within the action-perception cycle. Motor schemas define a subclass of schemas in that their primary function is one of controlling the activity of the actuators. Perceptual schemas acquire and process sensory information. Hybrid schemas combine perceptual and motor schemas in one unit which provides integrated sensing and acting. Action, therefore, refers to the modification of a property or relationship and sensing to the monitoring of properties or relationships. The actions and sensing performed by schemas includes modifying and monitoring the internal state of the system as well as the environment.

6.2.1 Structure

These schemas then need the following major components (see Figure 6.1): an activation mechanism, an event section, and a component for learning and adaptation. The activation mechanism monitors both the current goal state(s) of the system and part or all of the sensory input and activates or deactivates the schema based on preset conditions in both areas. The event section consists of a number of components which yield both parallel and serial activity, and comprises the "behavioral" section of the schema. Complex systems which deal with the ever changing world must be capable of adapting to new situations. The presentation here includes a "tracing component" (not necessarily disjoint from the others) to allow adaptation, or tuning, of the schema over time and

to provide a history of the particular instantiation of the schema.

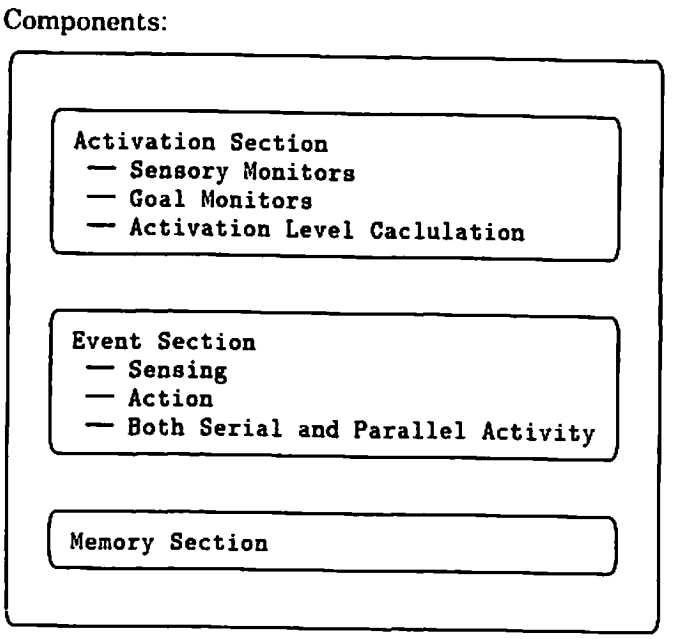


Figure 6.1

This diagram shows the three major components of a schema. The Activation Section turns the Event Section on and off according to the present situation. Sensory monitors and goal monitors are used to judge the fit of the schema to the present situation. The Event section specifies the action and sensing to be performed when the schema activates. The Memory Component, as discussed here, is intended as a place holder for an adaptation mechanism.

6.2.2 Activation Section

This structure will in some way code both the motor activity and sensory processing associated with some behavior. Such a framework must be capable of dealing with smooth transitions from one task to another with any number of tasks being partially active at any given time. Also of interest are the relationships between the action components of the control of a robot and objects in the world. The question arises: how do these units become active and how do they terminate their activity due either to completion of their piece of behavior or to some overriding external event. In short, how is it determined that the organism is in the correct domain of interaction for a schema to become active?

There are two possible activation paradigms: forced activation by a master scheduler, e.g. the Hearsay-II architecture [61], [125], or activation through a

distributed (competition and cooperation) scenario. In the first, a unit is needed for planning the activity of the organism and for decomposing that plan into the appropriate temporal sequence of activations. This requires that the scheduling unit contain models of the task, of the system which it controls, and of the environment, as well as specific knowledge regarding the details of the activation sequence. The use of a master scheduler sidesteps the question of activation knowledge within individual schemas by placing the control in the scheduler. The problem with this paradigm is the amount of knowledge which must be contained in a master scheduler. In some applications, a master scheduler may be appropriate, cf. the assembly schema below where direct activation of appropriately parameterized motor schemas is efficient and eliminates the need for the competition and cooperation inherent in distributed control. In the pseudo-AML formulation, this appears as a typical subprogram "call" with the usual parameter-passing conventions. However, it is the intent to develop methods that supplement more conventional programming strategies, and so the discussion here will concentrate on a distributed control strategy.

Distributed control can be accomplished by having all of the units continually monitor appropriate signal sets (which may include the sensory input, signals from other schemas, and the goal state of the system). A particular unit becomes active only when the sensory input and/or goals match the prerequisites for that unit or when directly activated by another unit. Each schema monitors at least one goal which calls for that schema's activation. To directly activate a schema, the goal specifically calling for its activation is added to the system's goal structure. Once an action has been initiated, temporal activation of additional units is typical. Distributed control eliminates the need for a single unit to possess knowledge of the entire world, goal state, and specific details of the way in which the organism should act. This is similar to the manner in which Demons in a "Pandemonium" system function. One of the differences, however, is that any number of schemas can be affecting the behavior of the system at a time.

This idea of sensory input-matching activating an "event" section of the schema may, at first pass, appear very similar to a production system. However, there are several differences. Recall that in a production system an exact match of the preconditions of a rule elicits immediate execution of the corresponding action. In schema activation, not only are the preconditions usually extremely complicated and parameterized, but the efficacy of the behavior dictated by the event section, e.g. movement of a given actuator, is determined by both the activation level of the schema and the set of other active schemas also providing output to that actuator. This is in contrast to the Pandemonium scenario in which a single Demon is active at a given time.

The activation section has two types of components: sensory monitors and goal monitors. Each sensory monitor (SM) receives input from the environment, or "internal sensory input" from other schemas, and has associated with it an output signal. The level of the output signal corresponds to how well the sensory input matches the characteristics of interest to the process. These characteristics

are held in the expected sensory input structures, see below, which are maintained by the DE. These are abstract types with elements whose structure fits the sensory modality being represented. The particular implementation of the elements is dependent on the programming language used to construct the DE and is not elaborated.

In a similar fashion, the goal monitors (GM) "watch" the system's goals and subgoals for particular patterns. As with the SM's, each goal monitor has an output signal which represents the fit of the expected goals with the actual goals. In the simplest case, the expected goal may be some particular event, e.g. part grasped, and the output signal may be binary. More complicated goals with explicit subgoal structures require output signals with continuously varying values which are dependent upon the partial completion of some set of the subgoals. The goal structure is also an abstract type and contains elements which describe desired states of the system. The DE maintains this structure and provides access facilities with which the schemas monitor the elements. As with the expected sensory input structures, the actual implementation of the structures depends on the programming language used and is not relevant here. Explicit definition of the goal monitors depends upon the ways in which goals are represented, e.g. symbolic or numeric, and the language used to implement the schemas and interface to the hardware, e.g. AML. In LISP, the goal structure could be a list structure where the depth of embedding of a particular goal encodes its global significance. In AML, the structure may appear as an aggregate of strings where each string is the name of a goal.

All of the outputs of the SM'S and the GM's are combined to produce the activation level of the schema. The method of combination, e.g. weighted average, sum, etc., is not of basic importance except for the fact that it must produce a graded signal which is indicative of the current fit of the sensory input and system goals to those expected.

Sensory monitors can be of varying complexity and are intended to encompass the general case of receiving sensory input and processing those data for a specific feature. For example, one sensory monitor could watch the strain gauge input from the fingertips of a robot gripper and "watch for" a particular force level.

A more complicated sensory monitor could be designed to detect a long thin object, using input from a tactile array sensor, see Chapter 4, E2. The monitor may require input from a single, specific sensor or from all available sensors. An algorithm which processes the tactile data is continually applied to the incoming sensory information. The processing algorithm could return an activity level expressing the confidence that a long thin object is being felt, together with parameters providing estimates for the centroid of the predominant region and the angle of the major axis.

These two examples illustrate the basic components of the sensory monitors. All SM's must specify the sensory modality and submodality which they cover as well as the somatotopic area (i.e. area of the "body" or sensory array) such as which strain gauge(s), tactile sensor(s), or part of a visual array

from which the sensory data are to be received. In addition, the type of processing must be defined, including the output parameters. Finally, a suitable processing algorithm must be supplied.

The specification of the modality and submodality allows certain assumptions, cf. Chapters 4 and 5, to be employed. Also, these specifications combined with the somatotopic area allow the particular sensor to be chosen. The processing algorithm may be as trivial as doing nothing when the raw sensory input is used directly in the output signal calculation, or it may be extremely complex and may, in fact, be represented by an entire hierarchy of schema activations. The output signal definition defines a manner in which the results of the processing, regardless of their abstraction, are used to produce a single output signal level. As with the goal monitors, the explicit specifications of the processing algorithm and output signal computation are dependent upon the details of the language used to access the hardware.

Once the schema activation level exceeds the predefined threshold for that schema, schema instantiation occurs. Two things happen at the time of instantiation: a trace begins and the event section is activated. An instantiation is a copy of the basic schema with particular values supplied for the various parameters in the basic schema. A number of instantiations of a given particular schema may exist, each with a particular set of values for the parameters. Two instantiations with identical values can exist simultaneously provided that they do not represent the same situation, i.e. object or interaction. The discussion now focuses on the event section and then briefly outlines the idea of the tracing, or memory, section.

6.2.3 Event Section

As mentioned earlier, the event section may contain interwoven activity of many actions with both serial and parallel activity. Activation of the event section causes simultaneous activation of the parallel components. Each of these components may, in turn, contain both serial and parallel activity. The structure of the components may be thought of as a graph, with the nodes representing points where decisions are made. Activity within the event section can take the form of actions or sensing. The "actions" may range from motor activity in the more straightforward case to the abstract action of modification of the system's goals. In many cases, a number of schemas require actions (either motor actions or goal state changes) to be performed by the same unit. A mechanism must be available to resolve such conflicts. One method could involve some form of competition or scheduling to allocate the resources among the involved schemas. The mechanism used here resolves such conflicts by making the ultimate action. or event, the combination of all of the requested actions weighted by the activation levels of their requesting schemas. When either the activation level drops below the threshold or the event section has completed, a "recovery" period is entered during which the activation level cannot be raised above the threshold. This "hysteresis" keeps a schema from being activated by the same SM's and GM's which initially activated the structure.

Communication between the units is accomplished via the event sections. Cooperation is accomplished when one unit's activity bolsters the activity level of another unit either by changing the other's goal state or providing evidence for the desired input through its own activities, or both. Note also that the actions performed by a schema may include the activation of other schemas, that is, establishing the goal and sensory configurations that will activate the new schema as well as defining the activity to be performed once the new schema activates. Competition occurs when one unit's activities either change the goal state away from that needed for another to activate or reduce the evidence for positive sensory input, or both.

6.2.4 Tracing Component

The discussion to this point has focused on the structure and activation of existing schemas. Two questions arise: how are schemas generated initially and how is a given schema tuned to more precisely fit a given situation? In most cases, the answer will be "because a human programmer (re)wrote the specification of the schemas" (with the assistance of the DE). However, it is maintained that a sophisticated robot control system will also include mechanisms for at least some automatic creation and tuning of schemas. To this end, it is proposed that schemas be provided with a tracing component which allows adaptation of the schema over time. Of course, in many practical applications, it will suffice to have schemas with only activation and event sections.

At every activation of a schema, the tracing component saves a copy of the sensory input and goal state at the instant of activation and begins temporal recording of the activity of the schema. Ideally, a continuous recording of activity, i.e. actuator states, would be made. However, in real systems, the time scale used is the finest available to the system. This trace represents a particular instance of a schema. Once completed, it provides the expected input for the SM's and GM's and the sequence of decisions and actions which describe the behavior of the event section. A copy of the original schema with these data in place of the original expectations becomes another schema. If the same activation parameters are encountered, this "tuned" version of the original schema is activated since it more precisely fits the data than the original, untuned version. As time progresses and this tuned version is not used, the time scale of the recording becomes larger; the parameters describing the expected input to the SM's and GM's of a tuned schema and the temporal activity of the behavioral section begin to lose definition. For example, you have a schema with an SM expecting an image of a cylindrical object. It becomes active. The axis of the cylinder is determined to have a particular angle, "x", and the value "x" becomes the expected value for the SM in the tuned schema. As time progresses and this tuned schema does not become active, the expected value "x" begins to lose definition, i.e. it becomes "x+-y" where the magnitude of y increases with time. When the expected values reach those of the original schema, the "tuned" version is no longer tuned and is removed from the system. The more often a

particular schema becomes active, the more resistant it becomes to this "forgetting" mechanism. One can think of this as functioning as if the decay constant becomes smaller with increased instances of activation.

The approach just outlined is distributed in that each schema contains a tracing component which is responsible for only that schema. In an alternate approach, the DE could handle the adaptation process. The DE would monitor which schemas become active, under what circumstances, and with what frequency. Several copies of a highly active schema could be tuned to slightly different situations. Tuning would increase the specificity of the sensory and goal monitors as well as restrict the set of possible actions. While the activation of schemas may best be handled by a competition and cooperation mechanism, a single mechanism which passively monitors the system may well be more effective for tuning purposes.

The important point here is not which, if either, of these mechanisms is employed, but rather that complex acting, perceiving systems must allow for adaptation. This allowance ranges from the simple tuning of an existing schema to situations requiring only slightly altered parameter settings as well as generalization of actions to entirely new situations. From merely outlining the three basic components, the discussion now focuses on a single example.

6.2.5 Example: *EMERGENCY*

A typical assembly task will be discussed in Section 6.3. Before getting to the details of the task, a particular schema will be presented to illustrate the general structure. In any robot system there must exist the ability to stop the machine in the event of a catastrophic failure. This "emergency stop" function brings all actuators to rest and usually removes power from them as well.

This function is served by the *EMERGENCY* Schema. The sensory monitors consider such quantities as the joint positions and velocities, motor currents, the water level around the robot's feet, temperature, smoke content in the air, etc. The single goal monitor expects the goal "stop robot." A schema can establish this system goal by an action in its event section. Existence of this goal causes the direct activation of the *EMERGENCY* schema. If any abnormal conditions arise which match those being monitored by the SM's, the schema activates fully, stops the machine and deletes all of the goals of the system. ("Delete" does not necessarily imply eradicating the goal structure from the system, but rather tagging the goals in such a way that they cannot be accessed by the schemas. Removing the structure would eliminate the possibility of restarting the system after maintenance.) Removal of the goals causes the activation levels of all other schemas to go below threshold, resulting in the full effect of the *EMERGENCY* schema's event section commands. Figure 6.2 shows the pseudocode of the *EMERGENCY* schema.

SCHEMA **EMERGENCY;**

```
-- Activation Section
      REPEAT_ALWAYS
          BEGIN
              activation_level = 0;
              CO_BEGIN
--
-- Sensory Monitors (SM's)
                  positions = INPUT_DATA (joint_positions);
                  velocities = INPUT_DATA (joint_velocities);
                  currents = INPUT_DATA (motor_currents);
                  forces = INPUT_DATA (finger_forces);
                  smoke = INPUT_DATA (smoke_level);
                  flood = INPUT_DATA (water_level);
--
-- Goal Monitors (GM's)
                  flag = CHECK_GOAL (stop_robot);
              CO_END
--
-- Activation Level Caclulation
              CO_BEGIN
                  IF OUT_OF_RANGE (positions(1)) OR
                      velocities(1) > max_velocity OR
                      currents(1) > max_current
                          THEN activation_level = 1;
                                  .
                                  .
                                  .
                  IF OUT_OF_RANGE (positions(number_of_joints)) OR
                      velocities(number_of_joints) > max_velocity OR
                      currents(number_of_joints) > max_current
                          THEN activation_level = 1;
                  IF forces(1) > max_force
                      THEN activation_level = 1;
                                  .
                                  .
                                  .
                  IF forces(number_of_finger_forces) > max_force
                  THEN activation_level = 1;
                  IF smoke THEN UPDATE (activation_level.smoke);
                  IF flood THEN UPDATE (activation_level.flood);
                  IF flag THEN activation_level = 1;
              CO_END;
          END;
      END_REPEAT;
```

```
-- Event Section
      DO_WHILE activation_level > 0
          CO_BEGIN
              SHUTDOWN;
              DELETE_GOALS;
          CO_END;
      END_DO;
END_SCHEMA;
```

Figure 6.2

Pseudocode for **EMERGENCY** Schema

The *EMERGENCY* schema utilizes two constructs for the control of the activation and event sections. The first of these is the

REPEAT_ALWAYS <statement>;

construct. This has the effect of causing the associated statement to be executed repeatedly ad infinitum. From a Computer Science perspective, this could be implemented as a process on a dedicated processor. Usually the statement will be a CO_BEGIN compound statement, see below. The statement begins again once it completes. This statement is used to control the activation section of the schemas. The sensory and goal monitors are processed and the activation level computed. In the ideal case, all monitors would be processed in parallel and the activation level updated continuously. On a processor network of normal architecture, the monitors would be evaluated in sequence and the activation level computed after each monitor was checked.

The second construct is the

DO_WHILE <logical expression> <statement>;

control structure which is used to implement control of the event section. In this case, the associated statement is executed only as long as the logical expression evaluates to TRUE. Usually the activation level computed by the activation section will appear in the logical expression. The activation section continues to monitor the sensory and goal input while the event section executes thus it is possible for the activation level to obtain a value which forces the

logical expression to evaluate to FALSE. In this case the DO_WHILE expression stops executing. As in the REPEAT_ALWAYS construct, the statement is usually a compound statement. If the statement finishes processing and the logical expression still evaluates to TRUE, then the DO_WHILE loop begins again. Neither the REPEAT_ALWAYS, as described here, nor the DO_WHILE statements are currently AML instructions.

The final construct of interest is a form of the standard compound statement. In this case a

```
CO_BEGIN
   S1;
   S2;
.
.
.
   Sn;
CO_END;
```

construct is used. Each of the statements S1, ..., Sn is executed simultaneously. Ideally, each function to be processed is provided with the computing resources needed whether that is a single processor or a set of processors.

The *CHECK_GOAL* schema simply monitors the goals of the system, continually checking for occurrence of the goal with which it was instantiated. Its sensory input is the goal structure and the action in the event section involves setting a flag. The implementation of the goal structure depends, as mentioned earlier, on the language being used to implement schemas.

The calculation of the activation level is a continual process as coded in the *EMERGENCY* schema. Basically, there are two types of contributions to the activity level. If a joint position moves out of range or a finger force exceeds a preset threshold, then the system must act immediately. Any element of this class of conditions causes the activation level to be set to 1 (the presentation here allows a range of [0,1] for activation levels) regardless of the contribution of other parameters. The other class of contributors is that with less immediate implications regarding the functioning of the robot. Each contributes to the activation level in an amount indicative of its presence in the environment: e.g. the smoke monitor contributes nothing if no smoke is present: a small amount if, say, cigarette smoke is present: and a larger value if dense smoke is sensed. Additions to the activity level for this class of elements is provided by the UPDATE function which takes the activation level and the contributing element as input. The function updates the activation level to include any change in the associated element. The effect is to subtract the previous value of the element from the activation level and add the new value. Thus the activation level is continually updated, in parallel, to include the state of all of the sensory and goal monitors.

6.3 Disk Assembly

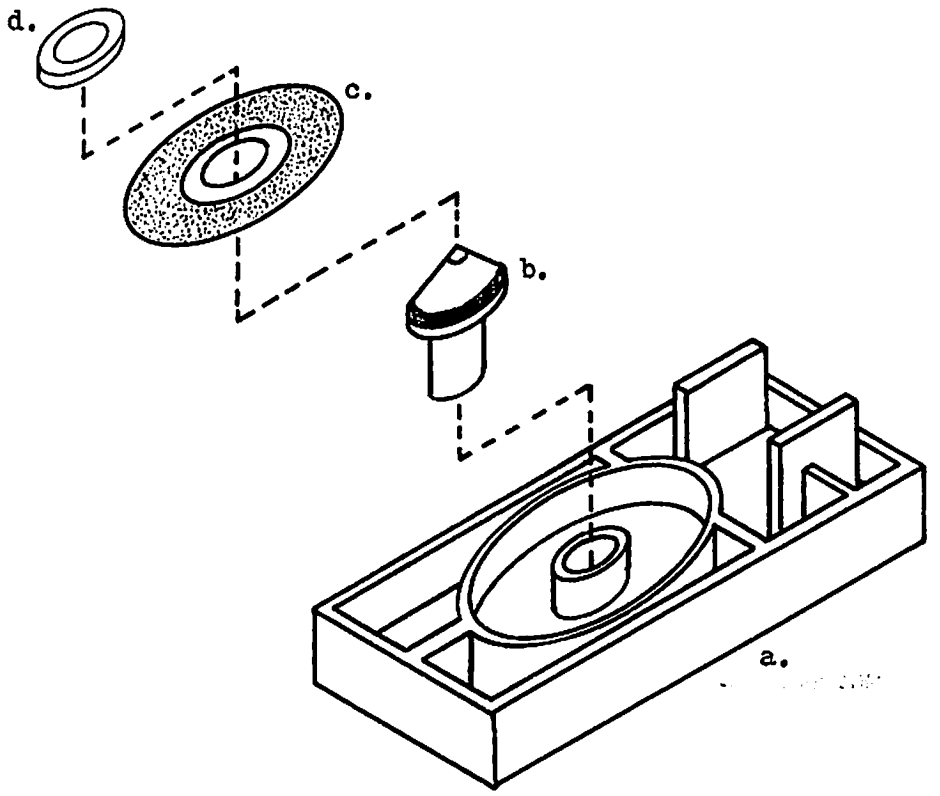
6.3.1 The Task

Let us now consider a possible robot task. The task involves one phase of the assembly of a Winchester technology disk drive: platter subassembly, combining the base-frame, a drive spindle, and the magnetically-coated aluminum platter, see Figure 6.3a. The assembly is generic in nature and as such does not include details particular to a given device. Figure 6.3b contains a pseudo flow diagram showing the steps necessary to complete the assembly. (The addition of the hold-down ring will not be discussed here.) This represents the gross flow of activity. Closer examination of this task reveals the amount of information and manipulative ability required. Machines do not have the sensory feedback or dexterity of humans; thus, all of these details must be specified to the robot.

The base-frame is a rectangular alloy casting on which all of the components of the drive are mounted. The front two-thirds of the frame is lower in profile than the rear section and has a shallow circular recess to accommodate the disk platter. In the center of the recess for the platter is an accurately bored hole to accept the spindle. Figure 6.3a contains a sketch of the base frame, part a. The first piece of information needed by the robot is the location and orientation of the base frame. In an assembly line with accurate mechanical feeders, it can be assumed that this information can be provided by the programmer as absolute positions in the world. However, in a more general situation, such data would need to be derived by processing the sensory information available to the robot. The locating phase must be repeated for each part to be manipulated. The robot must also have the ability to grasp and transport parts from their initial locations to the base frame.

The drive spindle is composed of two sections: a cylindrical bearing housing and a conical top section providing a seat for the disk platter. The bearing housing mates with the bored hole in the platter recess of the base-frame via a low-tolerance sliding fit. The upper section appears as a cone, the apex of which is machined to form a gear and is used to drive the platter. A groove is located near the base of the cone and provides a seat for the platter.

The spindle presents some problems not seen with the base-frame. Whereas the base-frame may be treated as fixed in space and right side up, the remaining parts are transported to the frame. The spindle thus needs to be moved from its initial location to the point where it is to be inserted into the base frame. This requires that its initial location be known as well as its desired final location relative to the hole in the base frame. The spindle can exist in any one of four stable configurations with two of the configurations allowing rolling. Thus the robot must not only determine the spindle location but also which stable configuration is present. From this, particular features can be chosen which, when their orientation/location is known, will provide the location and orientation of the piece. The features of interest in this case are: the ends of



6.3a

Figure 6.3

This figure illustrates the disk subassembly discussed in the text. 6.3a shows the four parts of the assembly: a. base-frame, b. spindle, c. disk platter, and d. the hold-down ring. 6.3b contains a diagram of the flow of activity during the assembly process.

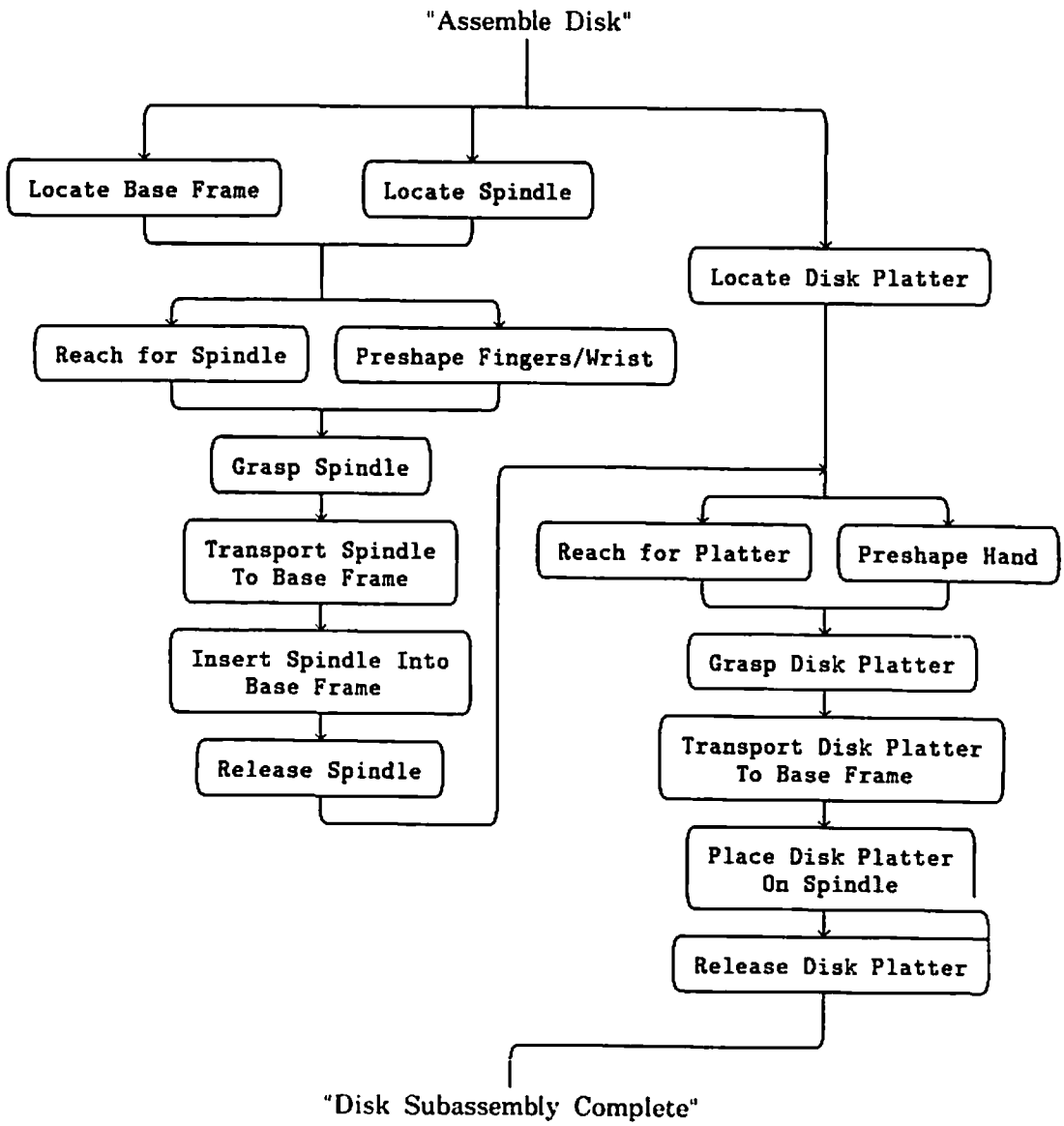


Figure 6.3b

Disk Assembly Steps

the cylindrical section and the edge of the base of the cone. It is known a priori how these features relate to one another. What is of interest here is their positions during the assembly. These features provide the information needed to grasp the object. Combining this information with the mass of the spindle, constraints on where the spindle can be touched, and how it mates with the base frame, allows the best grasp configuration to be determined and thus leads to the manipulation of the object. During the position recognition, grasping, and transportation phases, the only features of interest are those mentioned above. Dynamic tactile image processing becomes useful during the transportation phase since any change in the tactile sensation apart from that caused by the component of acceleration normal to the pad indicates that the part has changed positions.

The initial location and orientation of the spindle are required in order for the robot to obtain the part. When the spindle is in position to be inserted in the base frame, additional information is needed. The ends of the cylindrical section are again of interest as is the edge of the base of the conical section (since that is the grasping point) but of interest as well is the orientation and location of the surface of the cylindrical section. The insertion process can be viewed as the process of translating the spindle cylinder parallel to the axis of the hole while maintaining the separation of the hole wall and the spindle surface at a specified value. Once the insertion has been completed, the information regarding the cylindrical section of the spindle and the hole in frame is no longer needed. What is needed however, is the information regarding the conical section.

The disk platter itself is a circular aluminum plate. There is a hole in the center of the platter which mates with the groove in the drive spindle. Both surfaces of the disk are covered with a thin magnetic coating. The coating comes close to the hole in the disk and as it is used as the actual data storage medium, must be kept free from scratches, dust, grease, fingerprints, hydraulic fluid, etc. When coupled with the drive spindle, the platter is located radially by the groove in the drive spindle and secured in place by a hold-down ring and four screws.

The disk platter has only one stable configuration, lying flat, and there is no distinction between top and bottom. Since the disk is also symmetric about an axis through its center, no orientation information is needed. Thus the only relevant information concerns the location of the hole in the platter. Information regarding where the coating begins and ends radially relative to the hole is needed since the coating cannot be touched but is presumably in the static data base. As with the spindle, vision could be used to locate the platter and since it is planar, could be used to derive the location accurately enough that tactile or other sensory information would not be required.

When the platter is added to the assembly, the location of the apex of the groove in the conical section of the spindle must be known. The platter is grasped and transported to a position where an axis perpendicular to the plane of the platter through the center of the hole is roughly aligned with the spindle axis. In addition the plane of the platter needs to be roughly parallel to the

plane of the groove in the spindle. From this configuration the platter need only be lowered along the spindle axis to complete the assembly process. The conical shape of the spindle directs the platter during the phase when the parts are actually in contact with one another. Once the platter has been placed on the spindle, it may no longer be relevant to the assembly. That is, the assembly process could continue with only knowledge of the orientation of the axis and location of the apex of the conical section of the spindle, and the location and orientation of the hole in the base frame.

The pieces of information required fall into essentially two categories. The first contains invariant, pre-specified information provided as "global knowledge." For example, the axis of the cylindrical hole is known to be perpendicular to the plane of the gasket seat; thus, the axis orientation relative to the gasket plane need not be determined for each assembly. The second type includes variable information. While the relative orientation is static, the location of the spindle axis is not specified exactly from one assembly time to the next. Quantities in the latter class must be determined each time the assembly task is performed. The location and orientation of each part must be determined by the *LOCATE* schema, see below. Once determined, it can be stored with the part and only updated when things change or when unexpected events dictate that determination of the pose from scratch is required. Were all information to fall into the former class, acute sensing capabilities would not be required; and, in fact, a great deal of time and money is spent in industry to accomplish just this. Elaborate fixturing and parts presentation techniques are utilized to help reduce the amount of information which must be gathered by the system at the time of assembly.

We can immediately begin to see the components or primitives needed by such a system. The most obvious primitives involve the control of the actuators and sensors. High-level schemas require the ability to move the arm and sensors and to collect and process sensory data. For purposes of this discussion, we will assume that the manipulator is an IBM RS1 system with tactile array sensors on the fingertips and a camera mounted on an independent positioning system. The RS1 system is a six degree of freedom manipulator with a servo-controlled, two-finger gripper. The language AML is used to program these systems and an extended pseudo-AML is used here. The intent is to demonstrate how the concepts embodied in schemas may be implemented.

6.3.2 Acquiring The Drive Spindle

To gain some idea of what is really involved in one of these assemblies, the task of acquiring the drive spindle is considered in more detail. The first phase involves visually locating the spindle to provide the initial target location for a reaching movement. (In some cases, knowledge from the last assembly will include positional information which eliminates the need for a visual search.) During the reaching, the wrist is pre-oriented and the gripper is preshaped in a way which will afford grasping of the spindle. The grasp configuration is derived from perceptions of shape and orientation provided either from vision

or previous knowledge and, in general, the action to be performed. This preshaping places the fingers at approximately the proper separation and the wrist in roughly the correct orientation for grasping. Once the fingers contact the spindle, the exact position and orientation are determined from tactile sensation. The finger and wrist configurations are refined, and the spindle is grasped and then transported to the base frame.

The location phase employs one or more sensory modalities to scan the environment. In the case of vision, a region exhibiting the visual attributes associated with the spindle is sought. If the surface appearance is similar from one spindle to the next, the color and texture can be used as keys in the locating phase. The shape information can also be used. Here the system would search for the stimulus producing input patterns matching the projection of the spindle in one of its four stable configurations onto the imaging plane of the camera. If the particular spindle is unknown, e.g. a new model, a shape which can function as a spindle is sought, i.e. a shape which can fulfill the present goal of supporting a disk. Regions corresponding to objects of roughly the correct size and at approximately the correct distance are sought. Static vision provides information regarding the spectral attributes of objects while stereo vision and dynamic vision, e.g. optic flow, provide depth.

Once a set of candidate regions is found, a selection process is used to choose one. This process of choosing one region from the set will require additional sensory input in order to differentiate between the many possibilities. A more compelling example is the reaching and fondling activity needed to utilize the tactile sense. In the general case, algorithms for employing both contact and non-contact sensors, e.g. tactile and visual, must be available. This need for information may elicit motor activity to move various sensors around the environment. One example is the movement of the camera from one target to the next.

The approximate distance to the object is determined from stereo vision and size constancy, i.e. one expects the spindle to be of at least roughly known size. This knowledge, coupled with the known (via experience) parameters of the camera system, allows bounds to be established for the distance to the object. In order to locate a very familiar spindle, this process reduces to loading a stored input pattern, a mental image, and matching the present sensory input against that "image". Following location, the reaching, grasping, and transporting take place.

The reaching phase consists of a physical movement of the arm toward the spindle. Sensory information must be monitored in order to detect collisions and other unforeseen events. The distance and direction information is provided by the locating phase. The actual movement consists of two phases: a ballistic movement which puts the gripper in close proximity to the object, and a feedback phase in which visual and tactile sensation are used to bring the fingers into contact with the object.

Arm control is accomplished through an arm-move schema which again receives sensory input and goal input from other schemas and uses the lower

level motor control and sensory processing schemas to produce and adjust actions. This schema invokes, or activates, joint-move schemas with goals and parameter settings which result in the desired behavior. Smooth motor control of the arm actuators is the result of tuning all of the various schemas through experience. The arm-move schemas could invoke schemas above and below the joint level such as finger actuator schemas and wrist-move schemas. The ballistic movement is accomplished by activation of a position-control arm-move schema with the rough desired position determined elsewhere: from the locating phase in this case. The arm movements during the actual grasping of the spindle are controlled via coactivation of the force-control arm-move schema, force-control wrist-move schema, and the force-controlled grasp-schema for the appropriate configuration.

Grasping actually consists of two phases: grasp preshaping and final grasp under sensory control. The first phase consists of the preshaping of the fingers and wrist. Preshaping is performed during the reaching phase in order to prepare the hand for grasping the object. In the case at hand, the shape of the grasp is trivial since the RS1 employs a parallel-jaw gripper. The "shape" in this case refers to the finger separation and wrist orientation. In the case of a gripper with many degrees of freedom, the general shape is sculpted using the knowledge of the rough shape of the object. This phase prepares the gripper for tactile interaction with the object. The finger separation is maintained until sensory input is received which either triggers the final grasp phase or causes the separation to be altered. During the final grasp phase, the control mode may switch to force control.

In more complex tasks than the present one -- tasks using finely articulated hands -- once the fingers are in contact with the object, they are used to haptically search the surface, using tactile information so that a detailed model of the object can be constructed or verified. Grasping consists of applying forces with the fingers so as to constrain the motion of the object. It is certainly possible that the tactile sensory information received will differ significantly from that expected by the currently active hand configuring schema. In this case the configuration is modified by the currently active schema or another schema may become active while the activity level of the currently active schema declines.

The transportation schema moves the spindle from its initial location to the base frame. This schema uses the same arm-move schema that was used by the reaching schema. The difference between reaching and transporting is that the object must be accelerated and decelerated without slipping. The trajectory along which the spindle is moved is derived from the known kinematics of the robot and the locations of the spindle and base frame. Once grasped, the spindle's position relative the robot is known. The position of the base frame relative to the same robot-centered coordinate system is also known. The transportation schema utilizes information from the tactile sense, proprioceptive senses, and vision to provide feedback regarding the state of the spindle and the low-level motor schemas to move the arm.

6.4 Motor Schemas

The lowest level primitives allow movement of the arm, camera, and sensors and form the building blocks of higher-level schemas. An example of this is a joint control schema which determines the configuration and temporal characteristics of the joint which it controls. The joint control schema could implement any one of a number of control modes based on the current situation. In a position control mode, the configuration characteristic of interest is the angle of the joint and, temporally, the position should not change unless the desired position (as determined elsewhere) changes. Similarly, a velocity control mode could be used. Here the rotational velocity of the joint is of interest and the rate of change should be controlled. Finally, a force control mode could be used wherein the torque applied at the joint is of interest.

In each of these modes, sensory feedback is monitored by the joint control schema and used to adjust the action of the joint. In position control mode, the strain information from a primitive perceptual schema monitoring the tendon organs can be used to protect the system. If too large a force is encountered, the joint control schema could switch to the force control mode, signal an error condition, and change the position set point, or all of these.

This joint control schema clearly performs a servo-control function. The choice of the control mode is determined by the particular situation and the current goal for the system, i.e. what action is to be performed. The point to be made here is that even at the lowest level the schemas are combining both sensory information and the current goals to modify their functioning. This modification includes not only the typical servo functioning but also such "high-level" tasks as switching control modes.

The joint control schema as just described has the ability to choose and change control modes; in fact, there is actually a separate schema for each control mode. A particular control mode is entered when the schema implementing that mode becomes active. The activation can be due to global system goals or sensory information from the environment. Two control modes can be active simultaneously thus implementing, say, force control with a velocity sensitive component.

The next level of abstraction would provide finger-control schemas which coordinate the joint control schemas for the two fingers of the gripper. Again the control mode is a function of the action to be performed and the parameters necessary for each mode are determined from the situation and the sensory input. As with the joint control schemas, a separate but not necessarily disconnected schema would exist for each control mode. Again, the choice of control mode depends upon the current goal situation and the sensory input. The sensory input can be simple position data (from position or proximity sensors for instance) or complex, highly processed tactile array data. The set points for the control, i.e. the desired configuration of joint angles or velocities, are determined by the finger-control schemas again based on parameters and goals from higher-level schemas. A key point here is the coactivation and partial activation of multiple schemas. Not only are multiple schemas activated at any

given instant, but any particular actuator may be receiving input from a number of schemas simultaneously and a number of schemas may be receiving input from the same sensory process. The former property could be termed convergence since the mapping of "control generators" to actuators is not one-to-one. The mapping of a given sensory schema to many other schemas could similarly be termed divergence. Such coactivation can produce systems with variable gains in the control modes, e.g. stiffness in position control.

In the robot domain as constrained by AML, movements involving position can be either absolute or relative with limits placed on the velocity and maximum allowable force expected during the movement. A movement can also be specified by a direction vector and a velocity and stopping force. In this case, the manipulator moves along the direction vector at the prescribed velocity until either the specified force is felt at the gripper or one of the joints is requested to move out of its range. The primitive for an **absolute movement** is then:

AMOVE(device, position, max_velocity, max_force)

Here "device" is a pointer to the description of the mechanism to be moved. In our scenario this may include the manipulator, or a set of joints on the manipulator, the camera, or the tactile sensor(s). In the case of moving only a few of the joints on the manipulator, the remaining joints are locked. "Position" specifies the absolute position, in the coordinate system of the manipulator, to which the device is to be moved. The final two parameters specify maximums for the velocity and applied force tolerable during the movement. Note that there is an "affixment" relationship implied here analogous to that in languages such as AL. That is, if the tactile sensor is attached to the arm and the command *AMOVE*("tactile sensor",position) is issued, the arm will move as well. The function returns a value indicative of the success of the movement. Success occurs when the device reaches the requested position with neither the velocity nor force limits exceeded; failure is possible if either of the limits is reached or if the desired position is out of the legal working envelope of the manipulator. The following primitive handles **relative moves**:

RMOVE(device, d_position, max_velocity, max_force)

The parameters are defined as in the *AMOVE* primitive with the exception of the position specification: "d_position" contains the amount each of the joints is to be moved. This routine returns the status of the movement as in the *AMOVE* primitive. The final movement primitive provides **directed moves**, i.e. force-controlled movement along a given direction.

DMOVE(device, direction_vector, max_velocity, max_force)

The device is again defined as for the *AMOVE* primitive. The "direction-_vector" parameter specifies a direction in n-space, where "n" is the number of

joints, along which the device is to be moved. This primitive moves the device along the specified vector at a velocity specified by the third parameter until a force as specified by the final parameter is attained. This primitive can fail if the desired force is not attained and the device is moved to the edge of its working envelope. With these three primitives, all of the control modes are possible.

The *GRASP* schema is responsible for controlling the manipulator while acquiring an object. The schema requires information about the object to be grasped which includes the position and orientation and the places on the object which can be grasped. The intricacies of the grasping operation are too numerous to be detailed here; however the purpose of this schema is to grasp the object in the robot's gripper using the positional information determined by the *LOCATE* schema and information accessed through the **E_S_I_S**, see below. The latter information concerns both the sensory information to be expected during the interaction with the object as well as a simple program for the grasping of this particular object (from experience). While the previous commands were based on AML primitives, the *GRASP* schema is a new structure and is invoked directly by placing the "grasp" goal in the system goal structure. The activation appears as:

GRASP(object_features , position)

Here, "object_features" points to a structure which contains the features on the object which can be grasped and their positions and orientations relative in the object-based coordinate system. For instance, in the case of the spindle this structure would contain a feature known as a cylinder which would correspond to the body of the spindle. Along with this is the diameter and length of the cylinder and its orientation and position relative to the origin of the spindle coordinate system. Also included is a second cylinder corresponding to the edge at the base of the conical section. The conical section is not represented since it cannot be grasped with the two-fingered gripper. Also available to the *GRASP* schema is the system goal of assembling the disk drive. One result of this goal is that the spindle must be inserted into the base-frame which, in turn, implies that the cylindrical spindle body cannot be grasped (since it is the section that is to be inserted).

6.5 Perceptual Schemas

In our system, a camera and tactile array sensors are available in addition to the force sensors provided in each finger of the gripper. Two functions must be accomplished in regard to the sensors: data acquisition and processing. The camera provides an array of numbers corresponding to the amount of light sensed at each pixel; similarly, the tactile sensors provide arrays of numbers corresponding to the distribution of the forces and/or deflections present at the individual forcels in the pads. The raw patterns of sensory stimulation cannot guide movement directly. Rather, they must be interpreted using perceptual

schemas. The activation level of a schema encodes the credibility of the hypothesis that the object represented by the schema is indeed present; other schema parameters will represent further properties such as size, orientation, location, and relative motion which should be passed to the motor schemas.

INPUT_DATA(modality)

The above new primitive provides access to the sensor data: the primitive takes as a parameter the modality, or in this case the type of the sensor, and returns a data set from that sensor in the format appropriate for that device. Both the camera and tactile array sensor provide matrices of numbers. In the case of a single strain gauge, this primitive returns a single value. For each sensor, *INPUT_DATA* utilizes a lower-level schema which actually implements the data collection algorithms, e.g. the scanning algorithm used for the tactile sensor.

We recall from the discussion of the assembly task that the locating phase uses one or more sensory modalities, the set of expected sensory attributes, and the ability to move the sensors in order to determine the location and orientation of an object. At the highest level, then, the locator is invoked directly in our pseudo-AML as:

LOCATE(expected_sensory_input_structure, position)

Here, the first parameter is a structure which contains the expected sensory information for the object being sought. The second argument is the current best estimate of the position and is used as the starting point of the search. This argument may contain an estimate provided by another schema or the last location determined for the piece, or instance of the piece, or nothing. This estimate is updated by the locator through probing of the environment with various sensors.

The *LOCATE* schema employs a basic sensor utilization process. The strategy first utilizes the most distant sensors to provide a position estimate as refined as the sense allows, then repeats the process for each sensor. The estimate from one sense provides the starting point for the next sense and so forth until the object, as defined by expected patterns of sensory input, is: 1) found to the required accuracy, 2) not found at all, or 3) all of the sensory modalities have been exhausted. Vision, being non-contact and providing a global view of the environment, is started before touch which is useful only after contact with the object has been made. The senses are used sequentially in certain cases and concurrently when the domains overlap.

The sensory input structure contains the expected data for each modality. For binary vision. stored information consists of the attributes of a binary image of the particular item (e.g. the base-frame in this case) viewed from various positions. For this example, the information includes not only the binary images but statistics for the images (e.g. the area, number of holes, etc.) and a list of features from the image and the expected relationships among the features. In

addition, an "expectation value" is carried for each view. This value encodes the likelihood that this particular sensor data configuration will be encountered.

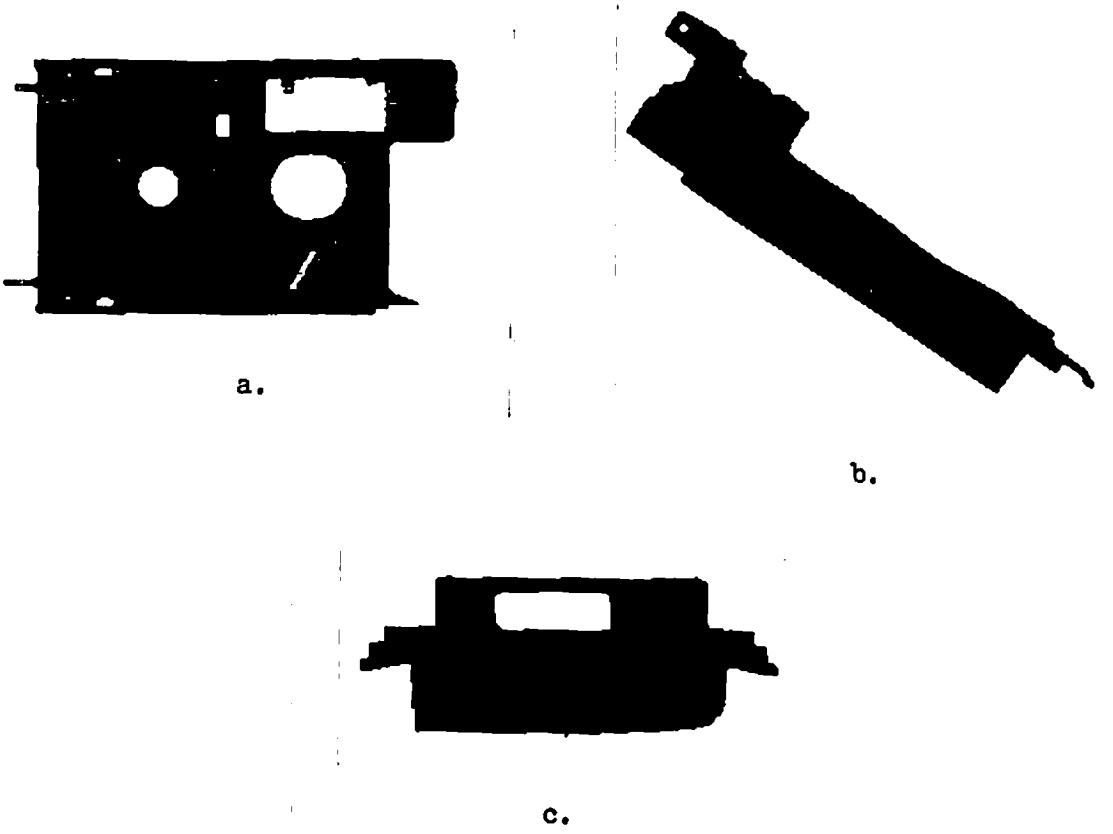


Figure 6.4

This figure contains three binary images of the base-frame component of the disk subassembly. The images were obtained using the SRI vision module which provides a 128x128 pixel image. These images were taken from the standard views of (a) top, (b) side, and (c) end.

For instance, the likelihood of seeing the base-frame standing on end when viewed from above is less than the likelihood for it standing on its side and even less than for it lying flat. In the case of tactile sensation, the location of tactile features (e.g. edges and corners) are stored, and these locations relative to some object-centered coordinate system are available. It should be noted that the sensory capabilities in the present system are rather primitive, i.e. binary visual image. In a more general robot system, the visual input is in the form of a grey-scale image and may also contain color information. For these data,

far more sophisticated processing techniques are employed, including the application of hierarchial processing cones as in the VISIONS system [92].

The information needed for the assembly task regarding the base frame consists of the location and orientation of the hole in the frame and the location of the plane of the top lip of the hole. The axis location and orientation are needed in order to align the cylindrical section of the spindle with the hole. We now discuss how this information may be obtained using binary images.

A set of standard views of the base-frame in each of its stable positions provides the basic set of vision expectations. Binary images of the base-frame viewed from the top, side, and end are provided in Figure 6.4a through c. Contained along with the images -- which may not exhibit the same scaling and rotation values as the actual sensory data -- are contained the statistics for the region in the image comprising the object. These statistics include the standard vector of characteristics used by the SRI or MIC vision modules: variable parameters such as the area and perimeter as well as invariant parameters such as the number of holes. Table 5 contains the feature vector values provided by the SRI Vision system for the images in the previous figure [4]. Other information, such as the relationship of features (edges, corners, and holes), may also be stored as part of the expected sensory input.

The structure of the ESIS (expected sensory input structure) can be depicted as in Figure 6.6. The main modality entries are ordered from most distant to least distant to the stimulus. Within each modality is an entry for each possible sensory input pattern (or set of patterns since, for example, a rotated image from a top view of the base-frame has essentially the same statistics and features as the standard view). Notice that the information contained about an object is that which is important in that modality and relates only to that modality. A common thread is woven through the various modalities by virtue of the fact that an object-centered coordinate frame is used; thus, in some sense, the features, or expected sensory input, in one modality can be "overlaid" on those from another.

The *LOCATE* schema moves from the distant sensors through the finest sensor, i.e. from the sensory with the least interaction with the object to the one with the most interaction. The starting location for the search in each modality is provided by the location algorithm applied to the previous modality. The position passed to the locator subroutine provides the initial starting point. In the case of a null initial position, a default value is used, e.g. it is begun wherever the sensor for the most distant sensor is sensing. From the starting point, the environment is scanned using the current sensor and a predetermined scanning strategy. Notice that this perceptual schema causes the robot to move, i.e. it employs motor schemas, thereby emphasizing the point that perceiving and acting cannot be differentiated.

In the case of the base-frame, vision and touch are the two senses. Vision is employed first with the initial image taken at the location specified by the "position" parameter in the activation of the primitive. An image is collected

Statistic	Top View	Side View	End View
Total area (pixels)	6,286	2,206	2,106
Area less holes	5,538	2,204	2,014
X maximum	100	117	119
X minimum	34	33	34
Y maximum	124	122	86
Y minimum	8	23	54
Area (image units)	5,538.0	2,204.0	2,014.0
Number of holes	7	1	1
Hole area (image units)	748.0	2.0	92.0
Total Area	6,286.0	2,206.0	2,106.0
Hole area ratio	0.1189	0.0009	0.0437
X center of gravity	66.84	78.92	74.42
Y center of gravity	62.04	73.00	69.39
Major Axis Length	108.82	115.02	80.63
Minor Axis Length	73.21	26.72	33.13
Major axis inclination	1.503	0.938	0.002
Axis ratio	0.6728	0.2323	0.4109
X perimeter length	142	200	170
Y perimeter length	262	214	70
Perimenter (image units)	384.083	312.659	223.598
Perimeter squared/ area	23.468	44.314	23.740
Number of boundary segments	68	346	56
Perimeter (pixels)	404	414	240
Sigma X	370,658	174,133	150,082
Sigma Y	344,083	161,100	139,955
Sigma X times Y	23,169,960	13,619,079	10,423,129
Sigma X squared	26,889,828	14,517,021	12,024,963
Sigma Y squared	2,164,631	1,091,760	821,403
Minimum radius (squared)	1,024	32	225
Maximum radius (squared)	4,624	4,717	1,940
Average (squared) radius	2,753.5	1,420.2	1,231.3
(Squared) radius ratio	0.2214	0.0068	0.1160
Angle of max radius vector	1.08084	-2.30469	-0.0454
Angle of min radius vector	3.14159	2.3562	1.5708
Difference of vector angles	-2.0608	1.622	-1.616
Length of bounding box	119.867	119.445	85.0156
Width of bounding box	72.1172	31.3203	32.2031
Length ratio	0.6016	0.2622	0.3788
Width in image units	66.0	84.0	85.0
Length in image units	116.0	99.0	32.0
Bounding box area	7,656.0	8,316.0	2,720.0
Box area - total area ratio	0.8211	0.2653	0.7743

Table 5

Table of feature vector values for previous images as determined by the SRI Vision module.

Expected_Sensory_Input_Structure

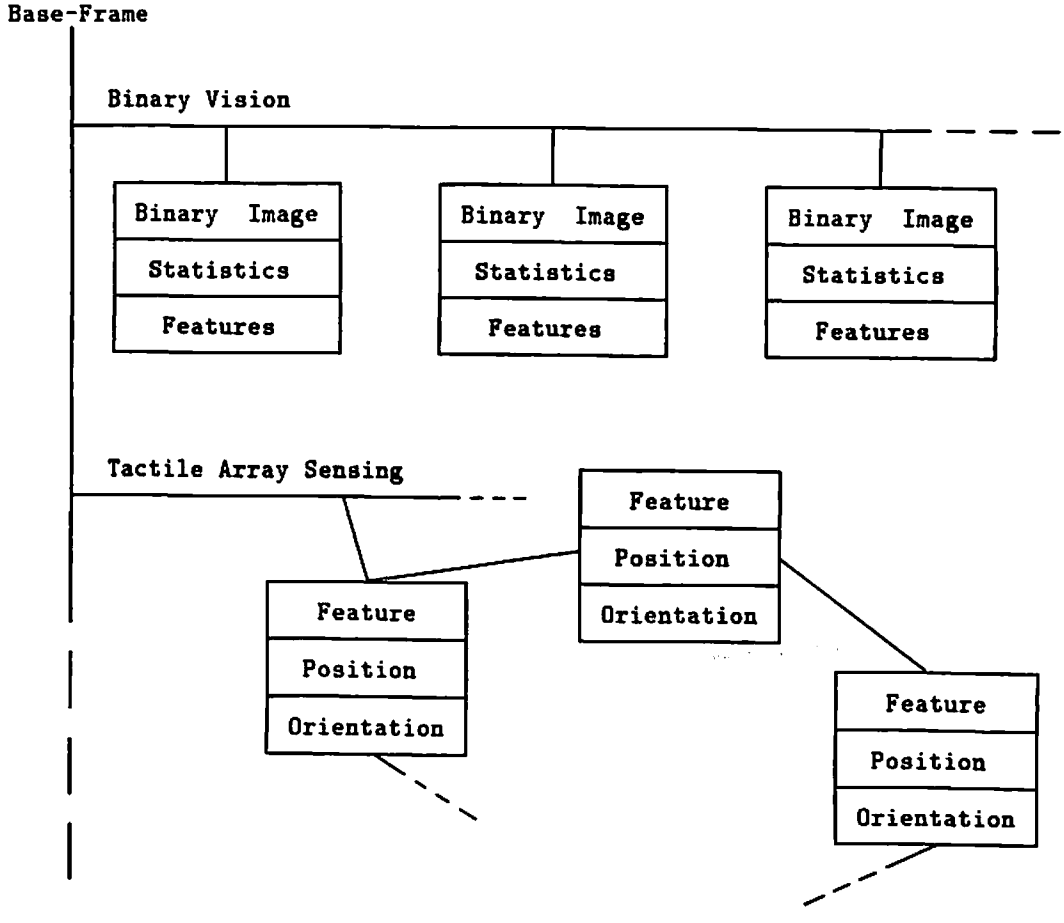


Figure 6.5

This figure depicts the structure of the ESIS. The main modality entries are ordered from most distant to closest to the stimulus. Within each modality is an entry for each possible sensory input pattern (or set of patterns since, for example, a rotated image from a top view of the base-frame has essentially the same statistics and features as the standard view). A common thread is woven through the various modalities by virtue of the fact that an object-centered coordinate frame is used to locate the features relative to one another.

and processed, and the results of the processing compared to each of the entries in the vision section of the expected sensory input structure. If no match is found, the camera is moved and the process repeated. Figure 6.6 contains a possible scanning path for a visual sensor.

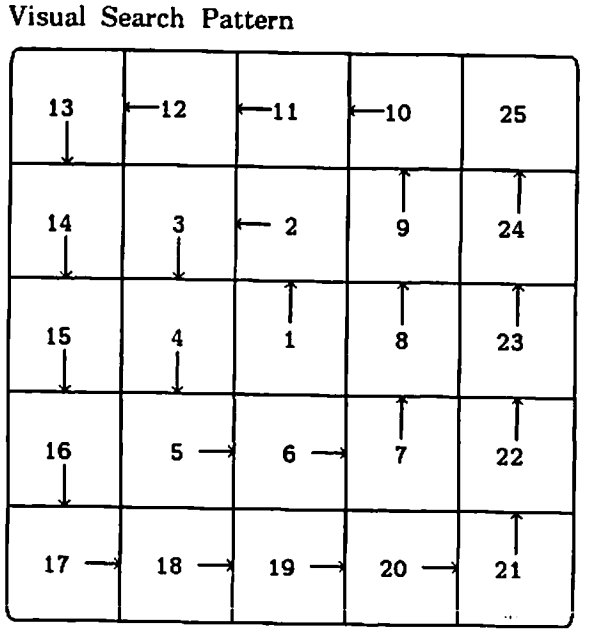


Figure 6.6

This figure shows a possible scanning path for a visual sensor. The squares are meant to represent images with the camera initially located to capture the image labeled 1. If, after this image has been compared with the expected visual input, it does not contain the desired stimulus, the camera is shifted so as to take the image in position 2. This process continues until either the stimulus is found or the search path has been exhausted.

This phase of the locating algorithm terminates when one of two conditions is met. Failure occurs when all of the sensor locations have been scanned and the desired sensory input not encountered; success results when the image taken in one of the positions matches one of the configurations in the input structure. Notice that the way in which the vision structure was defined allows three match techniques of increasing complexity. The simplest matching algorithm correlates the stored binary image with the raw input binary image. An exact match can only occur when the viewing parameters for the input data match those used when the expected stimulus input image was collected. Matching in this manner

will, typically, be a rare event. A second possibility for match exists with the feature vectors. Some of the features are scaling and rotation independent and these can be used to match the new image against an expected case. Finally, the relationships among the image features can be used by a method such as the Local Feature Focus method of Bolles [29]. What actually happens is that the input image is compared with all of the expectation using all three methods simultaneously. The "match" of the sensory data to the expected data is characterized by a confidence or "degree-of-match" criterion and the search stops when a value above a predetermined confidence value is obtained.

The locator gets the scanning algorithm from another data structure which contains an entry for each possible sensory modality. The scanning path is specified in a manner appropriate for the specific sensor. In the case of vision, a sequence of location relative to some starting point is given, cf. Figure 6.7. For touch, on the other hand, the definition of the scanning path is more complex. Here the *DMOVE* command would be used by a higher-level schema to produce a number of visually directed movements with the sensor until contact is made with an object. Once in contact, the scanning strategy changes to one of small positional changes in the plane of the sensor, then is followed by more probing.

As discussed, the revised position from a particular phase of the locator provides the initial input position for the next phase. This cycle continues, with the position refined at each stage until one of the following conditions results: the position of the object in question is determined to the required accuracy, or all of the sensory modalities have been processed, or failure is encountered in a modality. Notice that the locator is by no means a passive entity. On the contrary, the locator must scan the environment and this requires movement of the sensors. The example shown here contains relatively simple sensory input expectations; however, in more complicated situations, combined sensory input may be used, e.g. wrist force sensing for moment determination combined with shape determined from the finger joint positions and tactile sensor images.

The entire *LOCATE* schema is provided in pseudocode in Figure 6.7. The **E_S_I_S** may differ for each instantiation and the modality of the most removed sense in the structure is provided by the MODALITY function which interrogates the structure. Once a pointer to a modality has been established in this manner, it can be used to access the proper search strategy for that modality via Search_Strategies. The *MATCH* function compares the input from a given modality with that expected as defined by the **E_S_I_S**.

Note that the structure of the *LOCATE* schema discussed to this point is the event section general schema structure as defined in section 6.2. The activation mechanism is direct activation, accomplished by a goal monitor watching a simple activation state goal. If this goal is changed by another schema, then the *LOCATE* schema activates directly, i.e. direct activation. Along a more interesting line, the *LOCATE* schema monitors the information associated with each object and subassembly in its workspace. If the location and orientation of an object are unknown and the object will be manipulated, then the *LOCATE*

```
SCHEMA LOCATE (Expected_Sensory_Input_Structure,position,status);
--
-------------------------------------------------------------------
-- Activation Section
      REPEAT_ALWAYS
          CO_BEGIN
--
-- Sensory Monitors (SM's)
--
               data = INPUT_DATA ( MODALITY ( E_S_I_S(1) ) );
--
-- Goal Monitors (GM's)
--
               flag = CHECK_GOAL (locate_E_S_I_S);
--
-- Activation Level Caclulation
--
               activation_level = MAX (flag, MATCH (data,E_S_I_S));
          CO_END;
      END_REPEAT;
--
-------------------------------------------------------------------
-- Event Section
      DO_WHILE activation_level > 0.5
          BEGIN
               FOR i = 1,NUM_MODALITIES (E_S_I_S)
                    BEGIN
                         modality = MODALITY ( E_S_I_S(i) );
                         search_patern = Search_Strategies(modality);
                         modality_fit,position = APPLY ( search_patern, E_S_I_S(i) );
                         IF ~ modality_fit THEN
                              BEGIN
                                   status = 0;
                                   DELETE_GOAL (locate_E_S_I_S);
                              END;
                    END;
                    status = 1;
               END;
          END_DO;
      END_SCHEMA;
```

Figure 6.7

Pseudocode for the **LOCATE** Schema

schema will activate to find that location. If the activation level is low, the *LOCATE* schema's requests to move the arm (to use the tactile sensors) will have no influence on the manipulator's actual motions. If the system is not using the camera, the now activated *LOCATE* schema will direct the camera to gain the sensory information it needs. As the need for a part increases, the activation level of the particular instantiation for that part will also increase. At some point the need for the part, and therefore the activation level of the associated instantiation of the *LOCATE* schema, will reach a point where the *LOCATE* schema takes control of the manipulator. At the beginning of the assembly process the locations of both the base-frame and the spindle need to be determined. Two *LOCATE* schema instantiations, one for the base-frame and one for the spindle, are created. They compete for control of the arm until one instantiation's activation level increases above the other's due to a more dire need of the information, e.g. when one part is needed at an earlier stage of the assembly process than the other. This schema instantiation takes control of the manipulator, gains the information it needs and then terminates, leaving the manipulator available for the other instantiation(s).

6.6 Hybrid Schemas

Since the robot system has only the one task of disk assembly, there is only one high-level schema required. Thus competition and cooperation among the schemas do not occur except to a limited degree during the simultaneous locating of one part and manipulation of another. The *ASSEMBLE_DISK* schema is responsible for coordinating the activity involved in putting all of the pieces together. It must contain the plan for the assembly, i.e. the order in which the parts are assembled. From this order it must temporally coordinate the repeated "locate - reach - grasp - transport - insert" cycle until either all of the parts have been assembled or unexpected sensory input is received, indicating a problem. A problem can be handled by the assemble schema, or perhaps the activity level of the assemble schema could drop and that for a special-purpose trouble-shooting schema increase. The assembly operation is accomplished by the *ASSEMBLE_DISK* schema developed below. Typically, the robot system programmed for this task will be required to do only this task; thus, activation of the *ASSEMBLE_DISK* schema will be direct and competition and cooperation will occur only among the schemas associated with this high-level schema.

Associated with the expected sensory input structure for each part is the technique to be employed by the robot for mating of that part with the rest of the subassembly. Activation of the *ASSEMBLE_DISK* schema occurs directly through the establishing of a goal of the system to build a disk drive. Once activated, the schema directly activates the *LOCATE* schema once for each of the parts in the assembly. The parts are then added one at a time to the base frame until the assembly is complete, or the *EMERGENCY* schema has stopped the machine, or the *LOCATE* or *ADD_PART* schema instantiations have failed, see below. Figure 6.8 contains the AML pseudocode for the *ASSEMBLE_DISK* schema.

```
SCHEMA ASSEMBLE_DISK;
--
-------------------------------------------------------------------
-- Activation Section
      REPEAT_ALWAYS
          CO_BEGIN
--
-- Sensory Monitors (SM's)
--
--                --none--
--
-- Goal Monitors (GM's)
--
                flag = CHECK_GOAL(need_disk);
--
-- Activation Level Caclulation
--
                IF flag THEN activation_level = 1
                    ELSE activation_level = 0;
          CO_END;
      END_REPEAT;
--
-------------------------------------------------------------------
-- Event Section
      DO_WHILE activation_level =/= 0
          CO_BEGIN
              SET_GOAL (locate_base_frame, POSITION (base_frame), s);
              SET_GOAL (locate_spindle, POSITION (spindle), s);
              SET_GOAL (locate_platter, POSITION (platter), s);
              BEGIN
                  ADD_PART (base_frame,spindle);
                  ADD_PART (spindle,platter);
              END;
          CO_END;
      END_DO;
END_SCHEMA;
```

Figure 6.8

Pseudocode for **ASSEMBLE_DISK** Schema

```
SCHEMA ADD_PART (part1,part2);
--
-------------------------------------------------------------------
-- Activation Section
      REPEAT_ALWAYS
          CO_BEGIN
--
-- Sensory Monitors (SM's)
--
               position1 = INPUT_DATA ( POSITION(part1) );
               position2 = INPUT_DATA ( POSITION(part2) );
--
-- Goal Monitors (GM's)
--
               flag = CHECK_GOAL (add_part_part2);
--
-- Activation Level Caclulation
--
               IF DEFINED (position1) AND
                    DEFINED (position2) THEN activation_level = 1
                    ELSE activation_level = 0;
          CO_END;
      END_REPEAT;
--
-------------------------------------------------------------------
-- Event Section
      DO_WHILE activation_level =/= 0
          BEGIN
               GRASP (part2);
               TRANSPORT ( INPUT_DATA ( POSITION (part1) ) );
               MATE (part1,part2);
          END_DO;
      END_SCHEMA;
```

Figure 6.9

Pseudocode for **ADD_PART** Schema

The *ADD_PART* schema monitors the position of the parts for which it was instantiated and the system goal of adding those two parts. It activates when both parts have been located and there is a system goal of mating the parts. Figure 6.9 shows the pseudocode for this schema.

Note that there are two sensory monitors, one for the position of each of the parts. These monitors are concerned with internal information and return a defined value only when the position is known. That is, the process of inputting these values does not succeed until their position is known, i.e. the *LOCATE* schema has successfully determined the position. As with the other schemas, *ADD_PART* monitors a system goal which defines its own activation. It must be the case that not only are the positions of the parts known but also the *ASSEMBLE_DISK* schema which is responsible for the ordering of the subassemblies must be at a stage where parts are needed before a particular instantiation of the *ADD_PART* schema activates.

Instantiation and activation of the *ADD_PART* schema will, in turn, cause the *LOCATE* schema to be instantiated and, eventually, activated for each of the parts required by the particular instantiation of *ADD_PART*. This is accomplished through the querying of the position of each of the parts by *ADD_PART*. The *POSITION* function, schema, in Figure 6.9 checks the system's knowledge regarding the given part. If the location is unknown, the *POSITION* schema then instantiates the *LOCATE* schema for that part.

In the event section, three actions are performed. The first of these is the grasping of the second part. The *MOVE* command is exactly as defined above where the position is determined again from the pointer to the information about that part/object and is characterized relative to the position and orientation of part1. The final action, *MATE*, denotes the application of a motor routine for mating the parts which is associated with that part pair. Mating of the first two parts begins when the cylinder contacts (or ideally, passes without touching) the lip of the hole. Sensory information in the form of force feedback from the wrist is used by the insertion schema. A certain temporal force profile is expected, and deviation from this profile by the actual sensory input may indicate problems with the task.

The purpose of this chapter has been to examine some of the issues inherent in the problems of incorporating multimodal sensory feedback into the planning and control process for robots within the action-perception cycle. These were approached using a paradigm wherein a user develops sophisticated programs through the use of a development environment (DE). The DE provides a shell connecting the user to the robot hardware and sheltering him/her from having to deal with the implementation details. Behavior being composed of sensory perception combined with action is expressed within the DE in terms of schemas, goal structures, and expected sensory input structures.

We introduced the notion of a schema which consists of three sections: the activation, event, and tracing sections. The activation section monitors sensory input and system goals, and it has associated with it an activation level

corresponding to the degree of match of the current situation to that for which the schema is coded. The event section contains the overt behavior appropriate for the expected situation. A tracing component would build a tuned version of the schema from the particular parameter values at the time of activation. Objects in and of themselves do not enter into the schemas explicitly, but details about the objects and, more importantly, the ways they interact are of interest. Sensory stimulation patterns produced during interaction with the objects provide access to further actions to be performed with the object. The relationship of the robot to the world is represented by the set of schemas active at a given instant. The schemas used in the example vary in abstraction from very low-level servo control processes (implemented in hardware) to high-level concepts such as ASSEMBLE. Again, all of the schemas correspond to actions, sensory processing, or a combination of the two. These ideas were embodied in routines written in pseudocode which greatly extends the AML programming language available on the IBM RS1 and RS2 manipulators.

AML was chosen since it represents one of the most advanced programming environments currently available for robots. From a Computer Science point of view, it is a reasonable choice since it contains as primitives the basic motion commands needed to control the robot, incorporates a structured programming philosophy for software development, and utilizes reasonable data structures. Extensions were suggested primarily to the set of control structures available. The REPEAT_ALWAYS statement causes the associated statement to be executed continually. This statement can be a single statement, a standard compound statement, or a CO_BEGIN compound statement. This statement differs from the normal statement in that all statements within the CO_BEGIN statement are executed simultaneously. The final statement is the DO_WHILE statement which causes the associated statement to execute as long as the logical expression controlling the DO_WHILE is true. An important underlying characteristic of these statements is that whenever a new stream of processing is required, e.g. as is the case with the multiple statements in a CO_BEGIN statement, it is assumed that individual, dedicated processing is available. This structure can cause the number of processors to grow indefinitely, and this problem as well as the multi-processor communication problem was not addressed.

This chapter provides motivation for thinking about the requirements of planning and knowledge representation systems which are destined to control sensate machines in real environments. The schema ideas offer a framework in which sensing and action are integrated.

Conclusion

This thesis is divided into six chapters plus an introduction and this brief conclusion. Chapter 1 discussed robot senses and presented a classification scheme for sensory information and a taxonomy for sensors. Chapter 2 compared and contrasted other tactile array sensors with the device developed expressly for this work. The final version of the sensor contains an array of 128 force transducers in a one-square-inch area and produces tactile images containing 256 forcels after a preprocessing phase. An analysis of the tactile sensor and algorithms for removing inherent offsets and nonlinearities was provided in Chapter 3. Static tactile image processing was covered in Chapter 4; the issues surrounding dynamic tactile image processing were addressed in Chapter 5. Discussions covered the characteristics of the domain and their use in simplifying the algorithms. Chapter 6 presented a schema approach to the utilization of sensory information in control. The approach taken developed an interaction-based, distributed control structure which implemented the idea of coordinated sensing and acting.

Tactile array sensors are only now reaching the stage where they are reliable and available in great enough numbers to allow the study of robotics tactile perception. The purpose of this work has been to provide the ground work for future studies in the field; certainly, continued work is needed in all areas.

While the tactile sensor used here provided a useful vehicle for the study of robot tactile sensing, tactile array sensors are still not suitable for widespread use in industry. The sensor presented in Chapter 2 requires several aspects developed in order to become generally applicable. The most obvious area involves spatial resolution. For certain classes of tasks the resolution of the present device is sufficient; however, as robots are required to assume a greater role in a human world, their sensory capabilities will need to approach, and sometimes exceed, those of humans. Thus, increased spatial and force resolution will be needed in future devices. Though this sensor was fabricated on a rigid plate, the ability to conform a tactile sensor to an arbitrary shape would allow its application in a larger set of situations. In addition, the physical ruggedness, interconnect structure with the host, and fabrication costs must be improved. Nevertheless, these characteristics are of a superficial nature. The truly important areas for future work surround the issues of sensory processing and integration.

Chapter 4 provided an overview of the current state of tactile image processing. This chapter outlined the characteristics of the static tactile data domain and several experiments using algorithms based upon these characteristics. Algorithms for finding features in tactile images and identifying objects through touch were presented with the intention of providing a first step in understanding tactile image processing in the static case. All of the algorithms were implemented on a serial processor which was considered to be separate from the sensor. Many of the algorithms contained distributed, local computations designed to be implemented in parallel. Further work is needed to apply

the parallel architectures used in other areas of Artificial Intelligence to this domain. In addition, the work in shifting data processing onto the sensor with processing located at each transducer must be continued, as it is indicative of the way in which sensory data should be processed in future, highly complex systems.

Chapter 5 dealt with processing in the dynamic case, presenting experiments in tracking of objects moving across the sensor and in the construction of representations from multiple images. The approach taken was one of passive touch wherein a sequence of static images was collected and processed. The area of active touch involves using the tactile sense in a fashion tightly coupled with the control of the mechanism; this approach will become increasingly important. Work in this area will become possible as the sensor technologies, processing techniques, and computer architectures develop. The integration of kinesthetic sensing with tactile array sensing will also play an important role in the development of object representation structures.

Chapter 6 examined some of the issues inherent in the problems of incorporating multimodal sensory feedback into the planning and control process for robots within the action-perception cycle. The idea of a **schema** was introduced as the elemental building block encoding the robot behavior within a development environment. A schema consists of three sections: (1) the activation section, which monitors sensory input and system goals and has associated with it an activation level corresponding to the degree of match of the schema to the current situation, (2) the event section, which contains the actions and sensing appropriate for the expected situation, and (3) the tracing section, which provides a memory component of some form to afford tuning of the schema to a particular situation. The intent of this chapter was to provide motivation for thinking about the requirements of planning and knowledge representation systems which are destined to control sensate machines in real environments. The schema ideas offer a framework in which sensing and action are integrated.

This dissertation has addressed a number of issues in the area of tactile array sensing for robots. It is hoped that the work contained here will stimulate interest in this sensory modality and foster continued work.

Bibliography

[1] A Manufacturing Language Reference and Screen Editor, IBM 7565 Manufacturing System Software Library, International Business Machines, Boca Raton, FL, 1982.

[2] Abramowitz, J.D., Goodnow, J.W., and Paul, B., "Pennsylvania Articulated Mechanical Hand - An End Effector to Determine Shape by Touch," Proceedings of the ASME 1983 International Computers in Engineering Conference and Exhibit, Chicago, Illinois, August 7-11, 1983.

[3] ADV11-A, KWV11-A, AAV11-A, DRV11 User's Manual, Digital Equipment Corporation, Maynard, Massachusetts, April, 1977, order number EK-ADV11-OP-002.

[4] Agin, G.J. and McGhie, D.F., "Vision Module User's Guide," Artificial Intelligence Center, SRI Project 4391, SRI International, Menlo Park, CA, January, 1980.

[5] Agin, G., "Real Time Control of a Robot with a Hand-Held Camera," Proceedings of the 9th International Symposium and Exposition on Industrial Robots, pp. 233-246, Washington, D.C., March 1979.

[6] Agin, G.J., "An Experimental Vision System for Industrial Application," Proceedings of the 5th International Symposium on Industrual Robots, Chicago, Illinois, 1975.

[7] Agin, G.J. and Duda, R.O., "SRI Vision Research for Advanced Industrial Automation," Proceedings of the 2nd USA-Japan Computer Conference, pp. 113-117, 1975.

[8] Albus, J.S., McLeon, C.R., Barbera, A.J., Fitzgerald, M.L., "Hierarchical Control for Robots in an Automated Factory," Proceedings of the 13th International Symposium on Industrial Robots and Robots 7, vol. 2, April 17-21, Chicago, Illinois, 1983.

[9] Albus, J.S., Barbera, A.J., and Fitzgerald, M.L., "Hierarchical Control for Sensory Interactive Robots," Proceedings of the 12th International Symposium on Industrial Robots, Paris, 1982.

[10] Arbib, M.A., Lawton, D.T., and Overton, K.J., "Perceptual Systems for Robots," COINS Technical Report 83-24, Computer and Information Science Department, University of Massachusetts, Amherst, 1983.

[11] Arbib, M.A., "Perceptual Structures and Distributed Motor Control," in the Handbook of Physiology, Section on Neurophysiology, Volume III on Motor Control, V.B. Brooks (ed.), The American Physiological Society, 1980.

[12] Arbib, M.A., "Segmentation, Schemas, and Cooperative Computation," in Studies in Mathematical Biology, Part I. Cellular Behavior and the Development of Pattern, S. Levin (ed.), Math. Assoc. of America, 1978.

[13] Arbib, M. A., "Parallelism, Slides, Schemas, and Frames," in Systems: Approaches, Theories, Applications, W. E. Hartnett (ed.), pp. 27-43, D. Reidel Publ. Co., 1977.

[14] Arbib, M. A., "Artificial Intelligence and Brain Theory: Unities and Diversities," Ann. Biomed. Eng., vol. 3, pp. 238-274, 1975.

[15] Armbruster, K. et al., "A Very Fast Vision System for Recognizing Parts and Their Location and Orientation," Proceedings of the 9th International Symposium and Exposition on Industrial Robots, pp. 265-280, Washington, D.C., March 1979.

[16] Bajcsy, R., "Shape From Touch," Seminar and Workshop on Sensors for Robotics and Automation, June 8-9, Kingston, RI, 1983.

[17] Ballard, D.H. and Brown, C.M., Computer Vision, Prentice-Hall, 1981.

[18] Barbera, A.J., Albus, J.S., and Fitzgerald, M.L., "Hierarchial Control of Robots Using Microcomputers," Proceedings of the 9th International Symposium on Industrial Robots, Washington, D.C., March 13-15, 1979.

[19] Bardelli, R., Dario, P., DeRossi, D., and Pinotti, P.C., "Piezo- and Pyroelectric Polymer Skin-Like tactile Sensors for Robots and Prostheses," Proceedings of the 13th International Symposium on Industrial Robots and Robots 7, vol. 2, April 17-21, Chicago, Illinois, 1983.

[20] Barrow, H.G. and Tenenbaum, J.M., "Recovering Intrinsic Scene Characteristics from Images," in Computer Vision Systems, A.R. Hanson and E.M. Riseman (eds.), Academic Press, pp. 3-26, 1978.

[21] Bejczy, A.K., "Smart Sensors for Smart Hands," AIAA/NASA Conference on "Smart" Sensors, Hampten, VA, 1978.

[22] Bejczy, A.K., "Effect of Hand-Based Sensors on Manipulator Control Performance," Mechanism and Machine Theory, vol. 12, 1977.

[23] Binford, T.O., "A Survey of Model-Based Image Analysis Systems," International Journal of Robotcis Research, vol. 1, no. 1, Spring, 1982.

[24] Binford, T.O., Brooks, R.A., and Lowe, D.G., "Image Understanding Via Geometric Models," Proceedings of the 5th International Conference on Pattern Recognition, Miami, pp. 364-369, December 1980.

[25] Binford, T.O. et al., "Computer Integrated Assembly Systems," Progress Report to NSF, Covering Period March 15, 1974 to September 15, 1974, Stanford AI Lab, September 1974.

[26] Birk, J.R. and Kelley, R.B., "An Overview of Basic Research Needed to Advance the State of Knowledge in Robotics," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-11, no. 8, August, 1981.

[27] Birk, J. et al., "General Methods to Enable Robots with Vision to Acquire, Orient, and Transport Workpieces," 6th Report to NSF, August, 1980.

[28] Bobrow, D. G., and Collins, A., Representation and Understanding: Studies in Cognitive Science, Academic Press, 1975.

[29] Bolles, R.C. and Cain, R.A., "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," in A. Pugh (ed.), Robot Vision, Springer-Verlag, New York, 1983.

[30] Bolles, R.C., and Fischler, M.A., "A RANDSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data," Proceedings of the 7th International Joint Conference on Artificial Intelligence, August 24-28, Vancouver, B.C., Canada, 1981.

[31] Bolles, R.C., "Part Acquisition Using the SRI Vision Module," Third International Computer Software and Applications Conference, IEEE Computer Society, Chicago, November 5-8, 1979.

[32] Bolles, R.C. and Paul, R., "The use of Sensory Feedback in a Programmable Assembly System," Stanford Artificial Intelligence Project Memo No. 220, Stanford University 1973.

[33] Bollinger, J.G., "Using a Tactile Sensor to Guide a Robotic Welding Machine," Sensor Review, March, 1981.

[34] Briot, M., "The Utilization of an 'Artificial Skin' Sensor for the Identification of Solid Objects", Proceeding of the 9th International Symposium and Exposition on Industrial Robots, pp. 529-548 Washington, D.C. March 1979.

[35] Brooks, R.A., "Symbolic Reasoning Amoung 3-D Models and 2-D Images," Ph.D. Dissertation, Computer Science Department, Stanford University, Stanford, CA, report no. STAN-CS-81-861, June, 1981.

[36] Burns, J.B., Hanson, A.R., and Riseman, E.M., "Extracting Linear Features," to appear in Proceedings of the International Conference on Pattern Recognition, August, Montreal, Canada, 1984.

[37] Burt, P.J., "Stimulus Organizing Processes in Stereopsis and Motion Perception," Computer and Information Science Dept. Technical Report 76-15, University of Mass., September, 1976.

[38] Butler, W.P. (T.S.P. Preparer), "Technical Support Package on Transducer with a Sense of Touch," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, November, 1979.

[39] Carterette, E.C. and Friedman, M.P., editors: Handbook of Perception Volume VIB, Feeling and Hurting, Academic Press, New York, 1978.

[40] Catros, J.Y., "Use of Optical Reflectance Sensors in Robotics Applications," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 12, December, 1980.

[41] Catros, J.Y. et al., "Automatic Grasping Using Infrared Sensors," Proceedings of the 8th International Symposium on Industrial Robots and 4th International Conference on Robot Technology, Stuttgart, Germany, May 30 - June 1, 1978.

[42] Cauna, N. and Mannan, G., "The Structure of Human Digital Pacinian Corpuscles (Corpuscula Lamellosa) and its Functional Significance," J. of Anatomy, vol. 92, pp. 1-20, 1958.

[43] Chouchkov, Ch.N., "The Fine Structure of Encapsulated Receptors in Human Digital Glabrous Skin," J. of Anatomy, vol. 114, no. 1, pp. 25-33, 1973.

[44] Clot, J. and Falipou, J., "Realization D'otheses Pneumatiques Madulaires, Etude D'un Detecteur de Pressions Plantaires," Publication LAAS 1852, Centre National de la Recherche Scientifique, Laboratoire d'Automatique et d'Analyse des Systems, Toulouse, France, December 1978.

[45] Clot, J. and Falipou. J., "Projet Pilote SPARTACUS: Etude d'un Capteur Tactile (Peau Articificielle) Utilise Comme Detecteur D'efforts de Pression et de Glissement," Publication LAAS 1629, Centre National de la Recherche Scientifique, Laboratoire d'Automatique et d'Analyse des Systems, Toulouse, France, December 1977.

[46] Couna, N., "Nature and Functions of the Papillary Ridges of the Digital Skin," Anatomical Record, vol. 119, no. 4, pp. 449-468, 1954.

[47] Craik, K.J.W., The Nature of Explanation, Cambridge University Press, 1943.

[48] Crosnier, J.J., "Grasping Systems With Tactile Sense Using Optical Fibres," Developments in Robotics 1983, Rooks, B. (ed.), IFS Publications Ltd., Kempston, Bedford, England, 1983.

[49] Dalamangas, C.A., "Performance of Elastomeric Connectors When Exposed to Corrosive Environments," 12th Annual Connector Symposium, Electronic Connector Study Group, October 17-18, Cherry Hill, N.J., 1979.

[50] Dario, P., Domenici, C., Bardelli, R., DeRossi, D., and Pinotti, P.C., "Piezoelectric Polymers: New Sensor Materials for Robotic Applications," Proceedings of the 13th International Symposium on Industrial Robots and Robots 7, vol. 2, April 17-21, Chicago, IL., 1983.

[51] deBoor, C., A Practical Guide to Splines, Applied Mathematical Sciences, vol. 27, Springer-Verlag New York, Inc., New York, 1978.

[52] DeSalvo, G.J. and Swanson, J.A., "ANSYS Engineering Analysis System User's Manual," vol. 1, Swanson Analysis Systems, Inc., Houston, Pennsylvania, March, 1983.

[53] Dixon, J.K., Salazar, S., and Slagle, J.R., "Research on Tactile Sensors for an Intelligent Naval Robot," Proceedings of the 9th International Symposium and Exposition on Industrial Robots, pp. 507-518, Washington, D.C., March 1979.

[54] Drake, S.H., "Using Compliance Instead of Sensory Feedback for High Speed Robot Assembly," Manufacturing Engineering Transactions, North American Metalwork Research Conference, Proceedings of the 6th, pp. 64-70, 1978.

[55] Drake, S.H., Watson, P.C., and Simunovic, S.N., "High Speed Assembly of Precision Parts Using Compliance Instead of Sensory Feedback," Proceedings of 7th ISIR, Tokyo, 1977.

[56] Duda, R.O. and Hart, P.E., Pattern Classification and Scene Analysis, John Wiley and Sons, New York, 1973.

[57] Ejiri, M., Uno, T., Yoda, H., Goto, T., and Takeyasu, K., "A Prototype Intelligence Robot That Assembles Objects From Plan Drawings," IEEE Transactions on Computers, vol. C-21, no. 2, 1972.

[58] EMI Shielding Products Catalog, Data Sheet No. D-860, Tecknit EMI Shielding Division, Cranford, NJ.

[59] EMI Shielding Products Catalog, Data Sheet No. D-840, Tecknit EMI Shielding Division, Cranford, NJ.

[60] Engelberger, J.F., Robotics in Practice: Management and Applications of Industrial Robots, American Management Associates, 1980.

[61] Erman, L.D. and Lesser, V.R., "The Hearsay-II System: A Tutorial," Chapter 16 in Trends in Speech Recognition, W.A. Lea (ed.), Prentice-Hall, Englewood Cliffs, NJ, 1978.

[62] Espian, B., and Catros, J.Y., "Use of Optical Reflectance Sensors in Robotics Applications," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 12, December, 1980.

[63] Finkel, R. et al., "AL, A Programming System for Automation," Computer Science Dept. Report CS-456, Stanford University, November 1974.

[64] Fitch, H.L. and Turvey, M.T., "On the Control of Activity: Some Remarks from an Ecological Point of View," in Psychology of Behavior and Sport - 1977, D.M. Landers and R.W. Christina (eds.), Urbana, IL, Human Kinetics Publishers, pp. 3-35, 1978.

[65] Flatau, C.R., "Force Sensing in Robots and Manipulators," Proceedings of the Second International CISM-IFToMM Symposium on the Theory and Practice of Robots and Manipulators, Warsaw, Poland, September 14-17, 1976.

[66] Folchi, G.A. et al., "Computer Controlled Pneumatic Retractable Search Sensor," U.S. Patent #4001156, January, 1977.

[67] Folchi, G.A. et al., "Six Degree of Freedom Transducer for a Manipulator System," U.S. Patent #3948093, April, 1976.

[68] Franzen, O. and Torebjork, E., "Precision and Ambiguity in Coding Vibrotactile Information," in Sensory Functions of the Skin of Humans, ed. D.R. Kenshalo, Plenum Press, New York, pp. 39-62, 1979.

[69] Garrison, R.L. and Wang, S.S.W., "Pneumatic Touch Sensor," IBM Technical Disclosure Bulletin, vol. 16, no. 6, November, 1973.

[70] Gaston, P.C., "Robotic Tactile Recognition," Master's Thesis, Artificial Intelligence Laboratory, MIT, Cambridge, MA, January, 1983.

[71] Geldard, F.A., The Human Senses, John Wiley and Sons, Inc., New York, 1954.

[72] Gescheider, G.A. and Verrillo, R.T., "Vibrotactile Frequency Characteristics as Determined by Adaptation and Masking Procedures," in Sensory Functions of the Skin of Humans, D.R. Kenshalo (ed.), Plenum Press, New York, pp. 183-206, 1979.

[73] Geschke, C.C., "A Variable Capacitance Touch Sensor," Proceedings of the 5th International Conference on Artificial intelligence, Cambridge, MA, August, 1977.

[74] Gibson, J. J., "The Thoery of Affordances," in Perceiving, Acting and Knowing, R. E. Shaw and J. Bransford (eds.), Erlbaum, 1977.

[75] Gibson, J. J., The Senses Considered as Perceptual Systems, Allen and Unwin, 1966.

[76] Gibson, J. J., "Observations on Active Touch," Psychology Review, vol. 69, pp. 477, 1962.

[77] Gibson, J. J., "The optical expansion-pattern in aerial location," American Journal of Psychology, vol. 68, pp. 480-484, 1955.

[78] Gibson, J. J., The Perception of the Visual World, Houghton Mifflin and Co., Boston, 1950.

[79] Glazer, F., "Computing Optic Flow," Proceedings of the 7th International Joint Conference on Artificial Intelligence, August 24-28, Vancouver, B.C., Canada, 1981.

[80] Gleason, G.J. and Agin, G.J., "A Modular Vision System for Sensor-Controlled Manipulation and Inspection," Proceedings of the 9th International Symposium and Exposition on Industrial Robots, pp. 57-70, Washington, D.C., March, 1979.

[81] Gordon, G. (ed.), Active Touch, The Mechanism of Recognition of Objects by Manipulation, Pergamon Press, 1978.

[82] Goto, T., Takfyasu, K., and Inoyama, T., "Control Algorithm for Precision Insert Operation Robots," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 1, January, 1980.

[83] Goto, T., Inoyama, T., and Takeyasu, K., "Precise Insert Operation by Tactile Controlled Robot HI-T-HAND Expert-2," in Proceedings of 4th International Symposium on Industrial Robots, Tokyo, pp. 209-218, 1974.

[84] Goto, T., "Compact Packaging by Robot with Tactile Sensors," Proceedings of 2nd International Symposium on Industrial Robots, Chicago, 1972.

[85] Gregory, R.L., "On How So Little Information Controls So Much Behavior," in Towards a Theoretical Biology, 2: Sketches, C.H. Waddington (ed.), Edinburgh: Edinburgh Univ. Press, 1969.

[86] Hackwood, S., Beni, G., Hornak, L.A., Wolfe, R., and Nelson, T.J., "A Torque-Sensitive Tactile Array Sensor, "Robotics Research, vol. 2, no. 2, Summer, 1983.

[87] Hanafusa, H. and Asada, H., "A Robot Hand with Elastic Fingers and its Application to Assembly," IFAC Symposium, Tokyo, 1977.

[88] Hanafusa, H. and Asada, H., "An Adaptive Control of Robot Hand Equipped with Pneumatic Proximity Sensors", Proceedings of the 3rd Conference on Industrial Robot Technology and the 6th Internationl Symposium on Industrial Robots, University of Nottingham, UK. March 24-26, 1976.

[89] Hanson, A.R., Riseman, E.M., and Glazer, F.C., "Edge Relaxation and Boundary Continuity," in Consistent Labeling Problems in Pattern Recognition, R. Haralick (ed.), Plenum Press, 1980.

[90] Hanson, A.R., and Riseman, E.M., (eds.), Computer Vision Systems, Academic Press, 1978.

[91] Hanson, A.R., and Riseman, E.M., "Segmentation of Natural Scenes," in Computer Vision Systems, A.R. Hanson and E.M. Riseman (eds.), Academic Press, pp. 129-163, 1978.

[92] Hanson, A.R., and Riseman, E.M., "VISIONS: A Computer System for Interpreting Scenes," in Computer Vision Systems, A.R. Hanson and E.M. Riseman (eds.), Academic Press, pp. 303-333, 1978.

[93] Harmon, L.D., "Tactile Sensing for Robots," Seminar and Workshop on Sensors for Robotics and Automation, June 8-9, Kingston, RI, 1983.

[94] Harmon, L.D., "Automated Tactile Sensing," Robotics Research, vol. 1, no. 2, Summer, 1982.

[95] Harmon, L., "Touch Sensing Technology: A Review," presented at the Conference on Military and Space Applications of Robotics, held at the National Academy of Sciences, Washington, D.C., November 3-5, 1980.

[96] Heigenbotham, W.B. et al., "Visual Feedback Applied to Programmable Assembly Machines," Proceedings 2nd International Symposium on Industrial Robots, pp. 77-88, May, 1972.

[97] Hill, J.W. and Sword, A.J., "Manipulation Based on Sensor-Directed Control: An Integrated End Effector and Touch Sensing System," Proceedings of the 17th Annual Human Factor Society Convention, Washington, D.C., October, 1979.

[98] Hillis, W.D., "Active Touch Sensing," A.I. Memo 629, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, April, 1981.

[99] Hirose, S. and Umetani, Y., "Kinematic Control of Active Cord Mechanism with Tactile Sensors," Proceedings of the Second International CISM-IFToMM Symposium on the Theory and Practice of Robots and Manipulators, Warsaw, Poland, September 14-17, 1976.

[100] Holland, S., Rossol, L., and Ward, M., "CONSIGHT-I: A Vision Controlled Robot System for Transferring Parts from Belt Conveyors," General Motors Research Symposium "Computer Vision and Sensor-Based Robots," September, 1978.

[101] Holland, S.W., Olsztyn, J.T., Rossol, L., and Dewar, R., "A General-Purpose Visual Object Locator," General Motors Research Laboratory Report CS-181, April, 1976.

[102] Horn, B.K.P., "Understanding Image Intensities," Artificial Intelligence, 8, pp. 201-231, 1977.

[103] Horn, B.K.P., "Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View," in The Psychology of Computer Vision, P. Winston (ed.), McGraw-Hill, 1975.

[104] Inoue, H., "Tactile Pattern Recognition of Manipulating Objects," Electro-technical Laboratory, vol. 35, no. 3, pp. 96-102, March, 1971.

[105] Jain, R., Martin, W.N., and Aggarwal, J.K., "Extraction of Moving Object Images Through Change Detection," Sixth International Joint Conference on Artificial Intelligence, Tokyo, 1979.

[106] Johansson, R.S., "Tactile Afferent Units with Small and Well Demarcated Receptive Feilds in the Glabrous Skin Area of the Human Hand", in Sensory Functions of the Skin of Humans, D.R. Kenshalo (ed.), Plenum Press, New York, pp. 129-152, 1979.

[107] Johansson, R.S., "Tactile Sensibility in the Human Hand: Receptive Field Characteristics of Mechanoreceptive Units in the Glabrous Skin Area", Journal of Physiology, Vol. 281, pp. 101-123, 1978.

[108] Johnston, A.R., "Proximity Sensor Technology for Manipulator End Effectors," Proceedings of the 2nd Conference on Remotely Manned Systems," California Institute of Technology, Pasadena, CA, 1975.

[109] Johnston, A.R., "Infared Lasar Rangefinder," NASA New Technology report no. NPO-13460, Jet Propulsion Laboratory, August, 1973.

[110] Kak, A.C., "Depth Perception for Robots," to appear as a chapter in Handbook of Industrial Robotics, S. Nof (ed.), John Wiley and Sons, New York, also Purdue University School of Electrical Engineering report TR-EE 83-44, October, 1983.

[111] Kanade, T., "Recovery of the Three-Dimensional Shape of an Object from a Single View," CMU U-CS-79-153, Dept. of Computer Science, Carnegie-Mellon University, October, 1979.

[112] Kenshalo, D.R., " Phylogenetic Development of Feeling", in Handbook of Perception Volume VIB, Feeling and Hurting, Carterette and Friedman (eds.), Academic Press, New York, 1978.

[113] Kinoshita, G., Ohishi, S., and Yoshida, M., "Development and Realization of a Multi-Purpose Tactile Sensing Robot," Developments in Robotics 1983, B. Rooks (ed.), IFS Publications Ltd., Kempston, Bedford, England, 1983.

[114] Knibestol, M., "Stimulu-Response Functions of Rapidly Adapting Mechanoreceptors in the Human Glabrous Skin Area", Journal of Physiology, vol. 232, pp. 427-452, 1973.

[115] Koenigsberg, W.D., "Noncontact Distance Sensing Technology," Seminar and Workshop on Sensors for Robotics and Automation, June 8-9, Kingston, RI, 1983.

[116] Kribestol, M. and Vallbo, A.B., "Single Unit Analysis of Mechanoreceptor Activity from the Human Glabrous Skin", Acta Physiologica Scandinavico, vol. 80, pp. 178-195, 1970.

[117] Larcombe, M.H.E., "Carbon Fibre Tactile Sensors," Proceedings of the 1st International Conference on Robot Vision and Sensory Controls, April 1-3, Stratford-upon-Avon, UK, 1981.

[118] Larcombe, M.H.E., "Tactile Sensors, Sonar Sensors, and Parallax Sensors for Robot Applications," 3rd Conference on Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, U.K., March 24-26, 1976.

[119] Larcombe, M., "Tactile Perception for Robot Devices," Proceedings of the 1st Conference on Industrial Robot Technology, Nottingham University, U.K., 1973.

[120] Lawton, D.T., "Processing Dynamic Images from a Moving Sensor," Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts, Amherst, February, 1984.

[121] Lawton, D.T., "Optic Flow Field Motion and Processing Image Motion," Proceedings of the International Joint Conference on Artificial Intelligence-81, Vancouver, B.C., Canada, August, 1981.

[122] Lawton, D.T., "Constraint-Based Inference for Optic Flow," Proceedings of the 1st National Conference on Artificial Intelligence, Stanford University, August 18-21, 1980.

[123] Lee, D.N. and Lishman, J.R., "Visual Control of Locomotion," Scandinavian Journal Psychology, vol. 18, pp. 224-230, 1977.

[124] Lee, D.N., "Visual Information during Locomotion," in Perception: Essays in Honor of J. J. Gibson, R. B. McLeod and H. L. Pick, Jr. (eds.), pp.250-267, Cornell University Press, 1974.

[125] Lesser, V.R. and Erman, L.D., "A Retrospective View of the Hearsay-II Architecture," Proceedings of the International Joint Conference on Artificial Intelligence-77, Cambridge, MA, 1977.

[126] Lieberman, L.I. and Wesley, M.A., "AUTOPASS, An Automatic Programming System for Computer Controlled Mechanical Assembly," Report RC 5925 (#25635), IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1976.

[127] Loewenstein, W.R. and Rathkamp, R., "The Sites for Mechano-Electric Conversion in a Pacinian Corpuscle", Journal of General Physiology, vol. 41, no. 6, pp. 1245-1265, 1958.

[128] MacKay, D.M., "Cerebral Organization and the Conscious Control of Action," in Brain and Conscious Experience, J.C. Eccles (ed.), pp. 422-440, Springer-Verlag, 1966.

[129] Marik, V., "Algorithms of the Complex Tactile Information Processing," Proceedings of the International Joint Conference on Artificial Intelligence-81, Vancouver, B.C., Canada, August, 1981.

[130] Marr, D., "Representing Visual Information," in Computer Vision Systems, A.R. Hanson and E.M. Riseman (eds.), pp. 61-80, Academic Press, 1978.

[131] Marr, D. and Poggio, T., "Cooperative Computation of Stereo Disperity," Science, vol. 194, pp. 283-287, October, 1976.

[132] Marr, D., "Early Processing of Visual Information," Philosophical Transactions Royal Society B275, pp. 483-524, 1976.

[133] Matsushima, K., Yamamoto, M., and Onaka, H., "On the Tactile Sensor for a Moving Robot and its Application," Proceedings of the 7th International Symposium on Industrial Robots, October 19-21, Tokyo, Japan, pp. 569-576, 1977.

[134] MINC-11, Book 7: Working with MINC Devices, Software Distribution Center, Digital Equipment Corporation, Maynard, MA.

[135] Minsky, M. L., "A Framework for Representing Knowledge," in The Psychology of Computer Vision, Winston, P. H. (ed.), McGraw-Hill, pp. 211-277, 1975.

[136] Minsky, M.L., "Steps Toward Artificial Intelligence," Proceedings of IRE, vol. 49, pp. 8-30, 1961.

[137] Moravec, H.P., "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot, " Ph.D. dissertation, Stanford University, September, 1980, also published as Robot Rover Visual Navigation, UMI Research Press, Ann Arbor, MI, 1981.

[138] Moravec, H.P., "Rover Visual Obstacle Avoidance," Proceedings of the 7th International Joint Conference on Artificial Intelligence, August 24-28, Vancouver, B.C., Canada, 1981.

[139] Mujtaba, S. and Goldman, R.,"AL User's Manual." Stanford Artificial Intelligence Laboratory, Memo AIM-323, Computer Science Department, Stanford University, 1979.

[140] Nagel, R.N., VanderBrug, G.J., Albus, J.S., and Lowenfeld, E., "Experiments in Part Acquisition Using Robot Vision," SME Technical Paper MS79-784, Society of Manufacturing Engineers, Dearborn, MI, 1979.

[141] Nagin, P.A., Hanson, A.R., and Riseman, E.M., "Studies in Global and Local Histogram-Guided Relaxation Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-4, no. 3, May, 1982.

[142] Neisser, U., Cognition and Reality: Principles and Implications of Cognitive Psychology, W. H. Freeman and Company, 1976.

[143] Nevins, J.L. et al., "Exploratory Research in Industrial Assembly Part Mating," 5th and 6th Reports, C.S. Draper Laboratory, reports R-1111 and R-1276, 1980.

[144] Nevins, J.L. et al., "Computer Controlled Assembly," Scientific American, vol. 238, no. 2, pp. 62-74, February, 1978.

[145] Nevins, J.L. et al., "Exploratory Research in Industrial Modular Assembly," Reports 1 through 4 prepared for NSF by Draper Lab, June, 1973 to August, 1976.

[146] Nitzan, D., "Assessment of Robotic Sensors," presented at NSF Robotics Research Workshop, Newport, RI, April 15-17, 1980.

[147] Nitzan, D., "Robotics Automation at SRI," Proceedings of MIDCON/79, Chicago, IL, November 6-8, 1979.

[148] Nitzan, D., Brain, A.E., and Duda, R.O., "The Measurement and Use of Registered Reflections and Range Data in Scene Analysis," Proceedings IEEE, vol. 65, pp. 206-220, February, 1977.

[149] Nitzan, D. and Rosen, C.A., "Programmable Industrial Automation," IEEE Transactions on Computers, vol. C-25, no. 12, pp. 1259-1269, 1976.

[150] Nitzan, D., "Scene Analysis Using Range Data," Technical Note 69, Artificial Intelligence Center, SRI, August, 1972.

[151] Okada, T., "Development of an Optical Distance Sensor for Robots," Robotics Research, vol. 1, no. 4, Winter, 1982.

[152] Okada, T., "A Short-Range Finding Sensor for Manipulators," Bulletin of the Electrotechnical Laboratory, vol. 42, no. 6, 1978.

[153] Okada, T. and Tsuchiya, S., "Object recognition by Grasping," Pattern Recognition, vol. 9, pp. 111-118, 1977.

[154] Okada, T. and Tsuchiya, S., "Three-Demensional Pattern Recognition with an Artificial Hand," Bulletin of the Electrotechnical Laboratory, vol. 35, no. 3, pp. 334-345, 1971.

[155] Olsztyn, S.W. et al., "An Application of Computer Vision to a Simulated Assembly Task," Proceedings of the 1st International Joint Conference on Pattern Recognition, pp. 505-513, 1973.

[156] Overton, K.J. and Williams, T.D., "Tactile Sensation for Robots: Overview and Experience," Proceedings of the ASME 1983 International Computers in Engineering Conference and Exhibit, Chicago, IL, August 7-11, 1983.

[157] Overton, K.J., and Williams, T., "Tactile Sensation for Robots," Proceedings of the 7th International Joint Conference on Artificial Intelligence, August 24-28, Vancouver, B.C., Canada, 1981.

[158] Ozaki, H., Waku, S., Mohri, A., and Takata, M., "Pattern Recognition of a Grasped Object by Unit-Vector Distribution," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-12, no. 3, May/June, 1982.

[159] Page, C.J., Pugh, A., and Heginbotham, W.B., "Novel Techniques for Tactile Sensing in a Three-Dimensional Environment," 3rd Conference on Robot Technology and 6th International Symposium on Industrial Robots, University of Nottingham, March 24-26, 1976.

[160] Paul, R.P., Robot Manipulators: Mathematics, Programming, and Control, The MIT Press, Cambridge, MA, 1981.

[161] PDP-11 MACRO-11 Language Reference Manual, Digital Equipment Corporation, order number AA-V027A-TC, Maynard, MA, March, 1983.

[162] Perkins, W.A., "A Model-Based Vision System for Industrial Parts," IEEE Transactions on Computers, vol. C-27, pp. 126-143, February, 1978.

[163] Perkins, W.A., "Model-Based Vision System for Scenes Containing Multiple Parts," Fifth International Joint Conference on Artificial Intelligence, pp. 678-684, August, 1977. (also available as GM Reseach Lab Report GMR 2386, 1977)

[164] Petit, Julien and Galifret, Yves, "Sensory Coupling Function and the Mechanical Properties of the Skin," in Active Touch. The Mechanism of Recognition of Objects by Manipulation: a Multi-disciplinary Approach, pp. 19-27, G. Gordon (ed.), Oxford: Pergammon Press, 1978.

[165] Piccirillo, T.P., Dalamangas, C.A., and Hasan, R., "Conductive Rubber Connectors - Their Applications and Reliability," Proceedings of the 22nd Annual Holm Seminar on Electrical Contacts, September 21-23, 1976.

[166] Pipitone, F.J. and Marshal, T.G., "A Wide-field Scanning Triangulation Rangefinder for Machine Vision," Robotics Research, vol. 2, no. 1, Spring, 1983.

[167] Popplestone, R.J. et al., "Forming Models of Plane and Cylinder Faceted Bodies from Light Stripes," Proceedings of the 4th International Joint Conference on Artificial Intelligence, pp. 664-668, September, 1975.

[168] Prager, J.M, and Arbib, M.A., "Computing Optic Flow: The MATCH Algorithm and Predictions," Computer Vision, Graphics, and Image Processing, vol. 24, pp. 271-304, 1983.

[169] Prazdny, K., "Motion and Structure from Optical Flow," Sixth International Joint Conference on Artificial Intelligence, Tokyo, 1979.

[170] Price, K. and Reddy, R., "Change Detection and Analysis in Multispectral Images," Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, MA, pp. 619-625, 1977.

[171] "Progress in Tactile Sensor Development," Robotics Today, vol. 5, no. 3, June, 1983.

[172] Pugh, A., Robot Vision, Springer-Verlag, New York, 1983.

[173] Purbrick, J.A., "A Force Transducer Employing Conductive Silicone Rubber," Proceedings of the 1st International Conference on Robot Vision and Sensory Controls, April 1-3, Stratford-upon-Avon, UK, 1981.

[174] Quilliam, T. Andrew, "The Structure of Fingerprint Skin," in Active Touch. The Mechanism of Recognition of Objects by Manipulation: a Multi-disciplinary Approach, pp. 1-18, G. Gordon (ed.), Oxford: Pergammon Press, 1978.

[175] Raibert, M.H. and Tanner, J.E., "Design and Implementation of a VLSI Tactile Sensing Computer," Robotics Research, vol. 1, no. 3, 1982.

[176] Raibert, M.H., Tutorial on Robotics - In Principle and Practice, International Joint Conference on Artificial Intelligence-81, Vancouver, B.C., Canada, August, 1981.

[177] Rebman, J. and Trull, M.W., "A Robust Tactile Sensor for Robot Applications," Proceedings of the ASME 1983 International Computers in Engineering Conference and Exhibit, Chicago, IL, August 7-11, 1983.

[178] Reinhold, A.G. and Vanderbrub, G., "Robot Vision for Industry: The Autovision System," Robotics Age, pp. 22-28, Fall 1980.

[179] Rosen, C.A., "Machine Vision and Robotics: Industrial Requirements," SRI International Technical Note 174, November, 1978.

[180] Rosen, C.A. et al., "Machine Intelligence Research Applied to Industrial Automation," Eighth report, SRI International, August, 1978.

[181] Rosen, C.A. and Nitzan, D., "Use of Sensors in Programmable Automation," Computer, vol. 10, no. 12, pp. 12-23, December, 1977.

[182] Rosenfeld, A. and Kak, A,C., Digital Picture Processing, Academic Press, New York, 1976.

[183] RTI Force Sensing Wrist FS-B data sheet, Robot Technology Incorporated, Los Altos, CA.

[184] Salerno, C.M., "Compliant Tactile Sensor Pad System," Seminar and Workshop on Sensors for Robotics and Automation, June 8-9, Kingston, R.I., 1983.

[185] Sato, N., Heginbotham, W.B., and Pugh, A., "A Method for Three Dimensional Part Identification by Tactile Transducer," Proceedings of the 7th International Symposium on Industrial Robots, October 19-21, Tokyo, Japan, pp. 577-585, 1977.

[186] Sato, M., "Response of Pacinian Corpuscles to Sinusiodal Vibration," Journal of Physiology, vol. 159, pp. 391-409, 1961.

[187] Seltzer, D.S., "Use of Sensory Information for Improved Robot Learning," SME Technical Paper MS79-799, Society of Manufacturing Engineers, Dearborn, MI, 1979.

[188] Shirai, Y. and Suwa, M., "Recognition of Polyhedrons with a Rangefinder," Proceedings of the 2nd International Joint Conference on Artificial Intelligence, pp. 80-87, London, September, 1971.

[189] Snyder, W.E., and St. Clair, J., "Conductive Elastomers as Sensors for Industrial parts Handling Equipment," IEEE Transactions on Instrumentation and Measurement, vol. IM-27, no. 1, March, 1978.

[190] Sugihara, K., "Automatic Construction of Junction Dictionaries and Their Exploitation for the Analysis of Range Data," Sixth International Joint Conference on Artificial Intelligence, Tokyo, 1979.

[191] Sugiyama, K., Matsuzaki, A., Nabeshima, Y., and Hibino, Y., "Automatic Container Handling Device with Tactile Sensor," Proceedings of the 7th International Symposium on Industrial Robots, October 19-21, Tokyo, Japan, pp. 587-595, 1977.

[192] Szabo, T., Dalamangas, C.A., Hasan, R., and Piccirillo, T.P., "Conductive Elastomeric Connectors in Electronics," 10th Annual Connector Symposium, October 19-20, Cherry Hill, NJ, 1977.

[193] Takeda, S., "Study of Artificial Tactile Sensors for Shape Recognition -- Algorithm for Tactile Data Input," Proceedings of the 4th International Symposium on Industrial Robots, pp. 199-208, Tokyo, November, 1974.

[194] Takeyasa, K., Goto, T., and Inoyama, T., "Precision Insertion Control Robot and its Application," Transactions of the ASME, Journal of Engineering for Industry, November, 1976.

[195] Tamai, T., "Electrical Properties of Conductive Elastomer as Electrical Contact Material," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, vol. CHMT-5, no. 1, March, 1982.

[196] Taylor, R.H., "A Survey of Robot Programming Languages," presented at the Conference on Military and Space Applications of Robotics, held at the National Academy of Sciences, Washington, D.C., November 3-5, 1980.

[197] Tecknit Connectors Catalog, Data Sheet No. CEC-0601, Tecknit, Cranford, New Jersey.

[198] Tenenbaum, J.M. and Barrow, H.G., "Experiments in Interpretation-Guided Segmentation," Artificial Intelligence, 8, No. 3, pp. 241-274, 1977.

[199] Trevelyan, J.P., et al, "Techniques for Surface Representation and Adaptation in Automated Sheep Shearing," Proceedings of the 12th International Symposium on Industrial Robots, June, Paris, France, 1982.

[200] Trevelyan, J.P., "Software Techniques for Automated Sheep Shearing," Austrailian Computer Bulletin, March, 1982.

[201] Toda, H. and Masaki, I., "Kawasaki Vision System - Model 79A," Kawasaki Heavy Industries, Ltd., Robot Section, Nishi-Kobe Works, Matsumoto, Hazentani-Cho, Japan.

[202] Ueda, M., Iwata, K., and Shingu, H., "Tactile Sensors for an Industrial Robot to Detect Slip," Proceedings of the 2nd International Symposium on Industrial Robots, May 16-18, Chicago, IL., 1972.

[203] Ullman, S., The Interpretation of Visual Motion, Cambridge, Mass.: The MIT Press, 1979.

[204] "United States-Japan Trade Report," Subcommittee on Trade of the Committee on Ways and Means, WMCP: 96-68, U.S. House of Representitives, September 5, 1980.

[205] User's Guide to VAL, A Robot Programming and Control System, Version 12, Unimation, Inc., 1980.

[206] Vallbo, A.B. and Johansson, R.S., "The Tactile Sensory Innervation of the Glabrous Skin of the Human Hand," in Active Touch: The Mechanism of Recognition of Objects by Manipilation, G. Gordon (ed.), Pergamon Press, pp. 29-54, 1978.

[207] Vamos, T., Bathor, M. and Mero, L., "A Knowledge-Based Interactive Robot-Vision System," Sixth International Joint Conference on Artificial Intelligence, Tokyo, 1979.

[208] VanderBrug, G.J., Albus, J.S., and Barkmeyer, E., "A Vision System for Real Time Control of Robots," Proceedings of the 9th International Symposium and Exposition on Industrial Robots, pp. 213-323, Washington, D.C., March 1979.

[209] Verrillo, R.T., "Vibrotactile Sensitivity and the Frequency Response of the Pacinian Corpuscle," Psychonomic Science, vol. 4, pp. 135-136, 1966.

[210] Vierck, C., "The Somatosensory System," in Handbook of Behavioral Neurobiology, Volume 1 Sensory Integration, R.B. Masterson (ed.), Plenum Press, New York, 1978.

[211] Wang, S.S.W. and Will, P.M., "Sensors for Computer Controlled Mechanical Assembly," The Industrial Robot, vol. 5, no. 1, pp. 9-18, March 1978.

[212] Ward, M.R., Rossol, L., Holland, S.W., and Dewar, R., "CONSIGHT: A Paractical Vision-Based Robot Guidance System," General Motors Research Laboratories Publication GMR-2912, February 1979.

[213] Watson, P.C. and Drake, S.H., "Pedestal and Wrist Force Sensors for Automatic Assembly," Proceedings of the 5th International Symposium on Industrial Robots, pp. 501-511, Chicago, September 1975.

[214] Will, P.M., "Computer Controlled Mechanical Assembly," Proceedings on the 5th International Symposium on Industrial Robots, Chicago, 1975.

[215] Williams, T.D., "Computer Interpretation of a Dynamic Image from a Moving Vehicle," Ph.D Dissertation, Computer and Information Science Department, University of Massachusetts, Amherst, MA, May, 1981.

[216] Williams, T.D., "Depth from Camera Motion in a Real World Scene," IEEE Transactions on PAMI, vol. PAMI-2, no. 6, pp. 511-516, November, 1980.

[217] Wolfeld, J.A., "Time Control of a Robot Tactile Sensor," Thesis for the Master's of Science Degree, Moore School of Engineering, University of Pennsylvania, August, 1981.

[218] Wong, P.C., and Hudson, P.R.W., "The Austrailian Robotic Sheep Shearing Research Programme," Proceedings of the 13th International Symposium on Industrial Robots and Robots 7, vol. 2, April 17-21, Chicago, IL, 1983.

[219] Woodham, J.R., "Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from Image Intensity," Proceedings of SPIE 22nd Technical Symposium, vol. 155, San Diego, CA, August, 1978.

[220] Yashida, M. and Tsuji, S., "A Machine Vision for Complex Industrial Parts with Learning Capacity," Proceedings of the 4th International Joint Conference on Artificial Intelligence, pp. 819-826, 1975.

[221] York, B.W., Hanson, A.R., and Riseman, E.M., "3D Object Representation and Matching with B-Splines and Surface Patches," Proceedings of the International Joint Conference on Artificial Intelligence-81, Vancouver, B.C., Canada, 1981.

[222] York, B.W., "Shape Representation in Computer Vision," Ph.D. Thesis, COINS Dept., University of Massachusetts, Amherst, February, 1981.

[223] York, B.W., Hanson, A.R.. and Riseman, E.M.. "A Surface Representation for Computer Vision," Proceedings of the IEEE Workshop on Picture Data Description and Management, Asilomar, CA, August, 1980.

[224] Yoshikata, K., "A Microcomputer Controlled Tactile Gripper Sensor for Robot Manipulation and Effector Design," Proceedings of the ASME 1983 International Computers in Engineering Conference and Exhibit, Chicago, Illinois, August 7-11, 1983.

Appendix A

IBM/MINC Interface

This appendix describes the hardware and software comprising a 16-bit parallel interface implemented between an IBM RS1 robot system (presently the IBM 7665 system) and a DEC MINC system. The purpose of this link was to facilitate data transfer between the two machines. In particular, it was necessary that a program executing on one machine be capable of communicating with a program running on the other. A DRV11 parallel I/O unit was used on the DEC machine and the DI/DO ports used on the IBM side. The use of the DI/DO facilities greatly reduces the transfer rate due to the poling nature of the access by the computer; in fact, 80 msec were required to transfer one 16-bit word, including the handshaking protocol.

One output and one input control bit was used on each side of the interface. On the DRV11 side, the Control Status Register CSR0 bit was used as the Output Control Bit (OCB) and the REQA bit functioned as the Input Control Bit (ICB) [3]. DO bit 16 and DI bit 16 were used as the OCB and ICB, respectively, on the IBM side [1]. No error checking other than the flagging of attempted simultaneous writes was implemented. Check summing or other error detection/correction protocol could be implemented. However, given the short transmission distance and extra time required for such checking, no such capabilities were used.

A change of state of the control bit was used for the synchronization. In the output routines, the OCB is set and the routine waits until the ICB has set indicating that the other device is ready to receive data. Once the ICB has set, the OCB of the sending machine is reset and the number of words to be transferred placed on the outputs (i.e. into the output data buffer in the case of the DRV11 and onto DO bits 0-15 in the case of the IBM system). The sending machine then sets its OCB, indicating that the data are ready and waits while monitoring the ICB. When the ICB goes low, the receiving device has loaded the data. This cycle repeats once for each 16-bit word to be transferred.

The complete flow of control for the sending and receiving routines is presented in Figures A1 and A2 respectively. The software for the DRV11 is written in MACRO-11 [161] and runs on a MINC system using an 11/23 processor. Figure A3 contains the MACRO code listing for the **RS1RCV** routine which receives data sent by the IBM machine. Figure A4 contains the macro code for the **RS1SND** routine which functions to send data from the DEC MINC to the IBM. AML [1] was used for the IBM routines; code listings for the **SEND** and **RECEIVE** routines appear in Figures A5 and A6, respectively. Figure A7 contains the AML code which implements the functions used by the higher level **SEND** and **RECEIVE** routines.

The software provides a programming connection between the two machines, but unfortunately, the logic levels of the two machines differ. Each of the DI ports on the IBM are connected to +36 volts through a pull-up

resistor. The outputs of the DRV11 are not tristated and the true (actual +4 volts) output level of the DRV11 will not set these inputs. In order to interface the two devices, the output bits of the DRV11 (16 data bits and the OCB) are used to switch transistors, the collectors of which are connected to the DI points on the IBM. The output voltage swing of the DO ports was sufficient to trigger the DRV11 inputs, thus no conversion was needed. The complete schematic is presented in Figure A8. The IBM inverts the logic levels of its inputs and outputs. Thus the inversion introduced by the transistor configuration is interpreted correctly by the IBM. The DRV11 must complement the data it receives due to the IBM output inversion.

Figure A1

Flowchart for the SENDING routines. The version for the DEC MINC system is written in MACRO11 and uses a DRV11 programmable parallel I/O module while the IBM version is written in AML and uses the digital inputs, the DI group. Each machine uses one dedicated input control bit and one dedicated output control bit. On the DEC side these are the REQA and CSR0 bits respectively and on the IBM side they appear as DI16 and DO16 respectively.

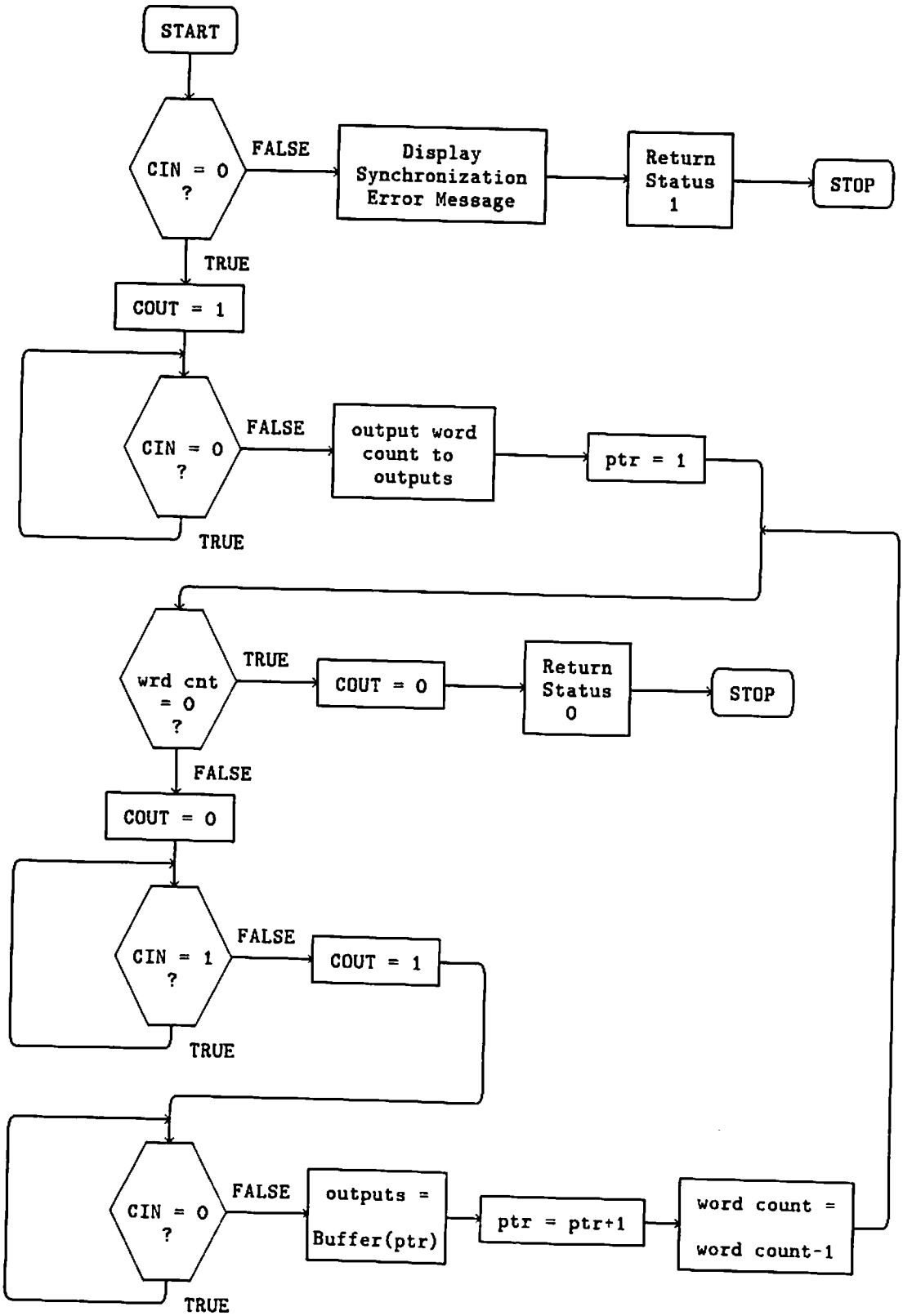
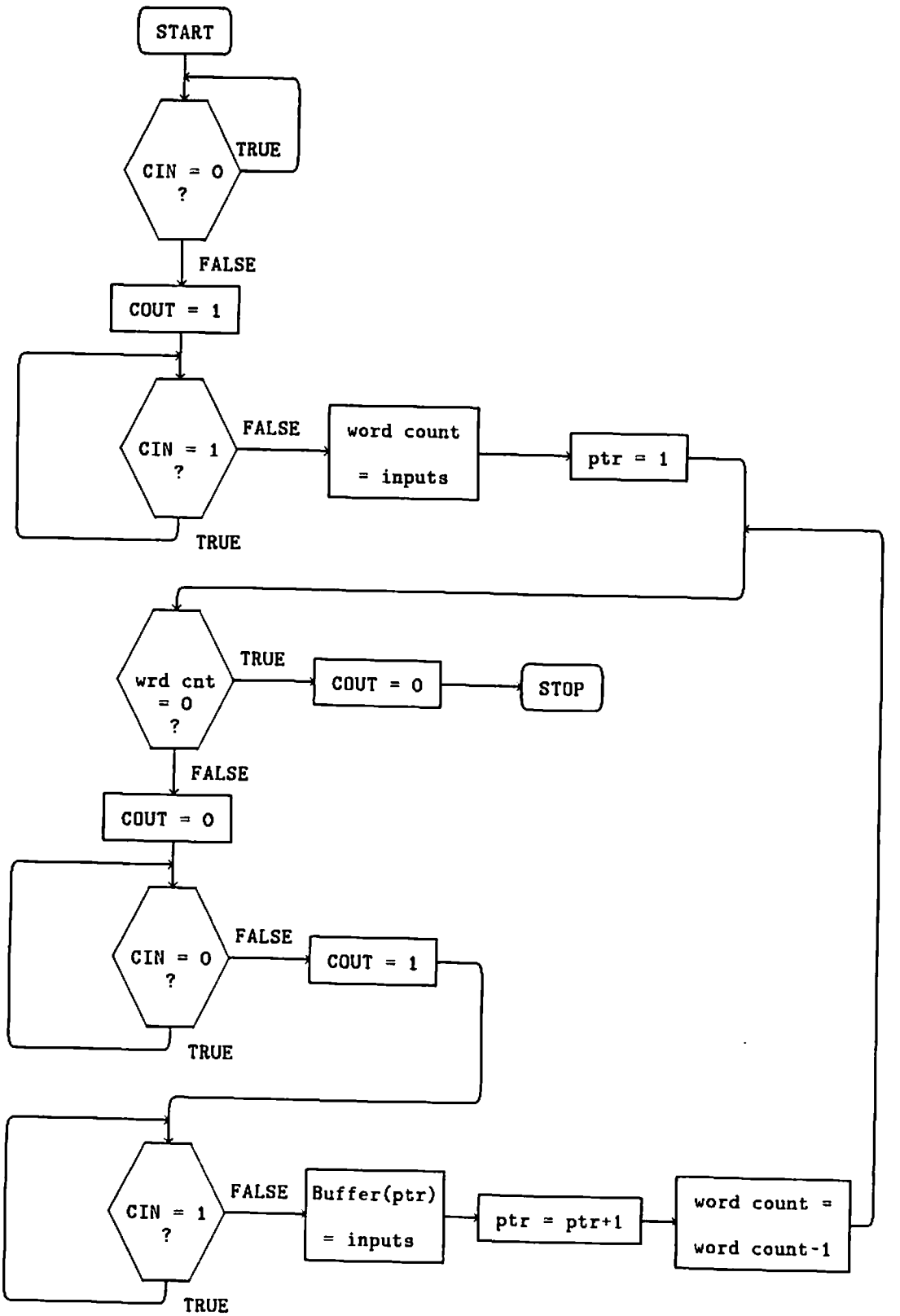

START
CIN = 0 ?
FALSE
Display Synchronization Error Message
Return Status 1
STOP
TRUE
COUT = 1
CIN = 0 ?
FALSE
output word count to outputs
ptr = 1
TRUE
wrd cnt = 0 ?
TRUE
COUT = 0
Return Status 0
STOP
FALSE
COUT = 0
CIN = 1 ?
FALSE
COUT = 1
TRUE
CIN = 0 ?
FALSE
outputs = Buffer(ptr)
ptr = ptr+1
word count = word count-1
TRUE

Figure A2

Flowchart for the RECEIVING routines. The version for the DEC MINC system is written in MACRO11 and uses a DRV11 programmable parallel I/O module while the IBM version is written in AML and uses the digital outputs, the DO group. Each machine uses one dedicated input control bit and one dedicated output control bit. On the DEC side these are the REQA and CSR0 bits respectively and on the IBM side they appear as DI16 and DO16 respectively.


START
CIN = 0 ?
TRUE
FALSE
COUT = 1
CIN = 1 ?
FALSE
word count = inputs
ptr = 1
TRUE
wrd cnt = 0 ?
TRUE
COUT = 0
STOP
FALSE
COUT = 0
CIN = 0 ?
FALSE
COUT = 1
TRUE
CIN = 1 ?
FALSE
Buffer(ptr) = inputs
ptr = ptr+1
word count = word count-1
TRUE

```
            .TITLE RS1RCV -- receive data from RS
 1
            .ENABLE LC
            .globl RS1RCV
;
;       This routine allows the transfer of data from the RS1 robot to the MINC via a
; DRV11 digital I/O module and the RS1 I/O ports.  The routine utilizes the DRV11
; Control Status Register CSR0 bit (CSR0) for an output control line and the Request
; A bit (REQA) for RS1 response.
;       The routine is accessed via:
;
;                   call RS1RCV(count,buffer)
;
;         where: count -- is the number of words to be transfered
;                buffer -- is the address of the start of the buffer
;                          containing the data
;
; address constant definitions
DRVCSR  =  176700                       ;DRV11 Control Status Register address
DRVIN   =  176704                       ;DRV11 Input Data Buffer Register address

RS1RCV:   tst       (r5)+               ;skip the arguement count

          mov       #0,@#DRVCSR         ;clear the DRV11 Control Status Register

wait1:    bit       #200,@#DRVCSR       ;check for request-to-send from the RS1
          beq       wait1               ;loop while waiting

          mov       #1,@#DRVCSR         ;set CSR0

wait2:    bit       #200,@#DRVCSR       ;check for reset of REQA
          bne       wait2               ;loop while waiting

          mov       @#DRVIN,r0          ;store word count in r0
          com       r0                  ;complement bits (IBM inverted outputs)
          mov       r0,@(r5)+           ;move word count to return location

          mov       (r5),r1             ;store the buffer address in r1
loop:     mov       #0,@#DRVCSR         ;clear control bit (CSR0)

          tst       r0                  ;check word count
          beq       return              ;branch to exit if wd cnt equal to 0

wait3:    bit       #200,@#DRVCSR       ;test for clear REQA (from RS1)
          beq       wait3               ;loop while waiting

          mov       #1,@#DRVCSR         ;set CSR0
                    - - - continued - - -
```

```
wait4:  bit     #200,@#DRVCSR   ;check for RS1 response (set REQA)
        bne     wait4           ;loop while waiting

        mov     @#DRVIN,r3      ;input the next word (store in r3)
        com     r3              ;invert the bits (IBM's inverted outputs)
        mov     r3,(r1)+        ;store number in return buffer

        dec     r0              ;decrement the word count
        jmp     loop            ;loop for next word

return: rts     pc              ;return to calling program
        .end
```

Figure A3

MACRO code for the **RS1RCV** routine which receives data sent from the IBM RS1 robot system to the MINC DRV11. The REQA bit of the Control Status Register is used as the input control bit and the CSR0 bit as the output control bit. These bits provide synchronization with the IBM.

```
            .TITLE RS1SND -- send data from the minc to the rs1 robot
            .ENABLE LC
            .globl RS1SND
;
;       This routine allows the transfer of data from the MINC to the RS1
; robot via a DRV11 digital I/O module and the RS1 I/O ports.  The routine
; utilizes the DRV11 Control Status Register CSR0 bit (CSR0) for an output
; control line and the Request A bit (REQA) for RS1 response.  The value of
; "status" upon return indicates the success or failure of the routine.
;       The routine is accessed via:
;
;               call RS1SND(count,buffer,status)
;
;       where: count -- is the number of words to be transfered
;              buffer -- is the address of the start of the buffer
;                        containing the data
;              status -- return value of 0 indicates successfully completed
;                        transfer, return value of 1 indicates that the REQA
;                        bit was sent at entry, i.e. the RS1 was (is) trying
;                        to send information at the same time
;
; address constant definitions
DRVCSR  = 176700                    ;DRV11 Control Status Register address
DRVOUT  = 176702                    ;DRV11 Input Data Buffer Register address

RS1SND: tst     (r5)+               ;skip the arguement count

        mov     #0,@#DRVCSR         ;clear the DRV11 Control Status Register

        mov     @(r5)+,r0           ;store word count in r0
        mov     (r5)+,r2            ;move address of buffer into r2

        bit     #200,@#DRVCSR       ;test for RS1 trying to send, return if T
        beq     skip                ;continue if NOT set

        mov     #1,@(r5)            ;return status of 1
        jmp     return              ;return from routine

skip:   mov     #1,@#DRVCSR         ;set CSR0

wait1:  bit     #200,@#DRVCSR       ;test for RS1 response (REQA)
        beq     wait1               ;loop while waiting

        mov     r0,@#DRVOUT         ;load the word count on the output port

loop:   mov     #0,@#DRVCSR         ;clear control bit
                - - - continued - - -
```

```
wait2:  bit     #200,@#DRVCSR   ;test for RS1 response (REQA)
        bne     wait2           ;loop while waiting

        tst     r0              ;check word count
        beq     return          ;branch to exit if wd cnt equal to 0

wait3:  bit     #200,@#DRVCSR   ;test for RS1 response (REQA)
        beq     wait3           ;loop while waiting

        mov     (r2)+,@#DRVOUT  ;load the next word on the outputs

        dec     r0              ;decrement the word count

        jmp     loop            ;loop for next word

        mov     #0,@(r5)        ;return status of 0 (success)
return: rts     pc              ;return to calling program
        .end
```

Figure A4

MACRO code for the **RS1SND** routine which sends data from the MINC DRV11 to the IBM RS1 robot system.

```
-- + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
-- "SEND" data from the RS1 to the MINC DRV11:
--          where:  Count -- (type INT) contains number of words to transfer
--                  Buffer -- (type AGGREGATE of INT) contains words to be sent
--          the return status -- indicates the success (0) or failure (1) of
transfer
SEND: SUBR(Count,!Buffer);
          ptr: NEW 0; -- index pointer for buffer
--
-- test the input control bit (ICB), if set the the DRV11 is
-- trying to send something, return immediately with status = 1
          IF CONTROL_IN NE 0 THEN
                  BEGIN
                   display(' * * * synchronization error * * *',EOL,EOL);
                   return(1);
                  END;
--
-- set the output control bit (OCB)
          CONTROL_ON;
--
-- wait for DRV11 logical true on the input control bit (ICB)
          WHILE CONTROL_IN EQ 0 DO 0;
--
-- output the word count to the output port
          SENSIO(OUT_BUF_LIO,Count);
--
-- loop while there are words left to send (i.e. Count =/= 0)
          WHILE (Count = Count-1) GE 0 DO
                  BEGIN
-- reset the OCB
                   CONTROL_OFF;
-- wait for ICB to reset
                   WHILE CONTROL_IN EQ -1 DO 0;
-- set OCB
                   CONTROL_ON;
-- wait for ICB to set (i.e. logical 1, 5 volts)
                   WHILE CONTROL_IN EQ 0 DO 0;
-- output the next word
                   SENSIO(OUT_BUF_LIO,Buffer(ptr = ptr + 1));
          END; -- while loop
-- clear the output control bit. return status = 0 for success
          CONTROL_OFF;
RETURN(0);
END; -- routine SEND
```

Figure A5

AML code for the **SEND** routine which transmits data from the IBM RS1 robot system to the MINC DRV11.

```
- + + + + + + + + + + + + + + + + + + + + + + + + + + + +
- "RECEIVE" data from the DRV11
- arguements as in routine SEND
-
RECEIVE: SUBR(!Count,!Buffer);
          tcount: NEW INT;
          ptr: NEW 0;
-- wait for ICB to set
          WHILE CONTROL_IN EQ 0 DO 0;
- set OCB
          CONTROL_ON;
-- wait for ICB to reset
          WHILE CONTROL_IN EQ -1 DO 0;
-- input word count
          tcount = SENSIO(IN_BUF_LIO,INT);
          Count = tcount;
- loop while word count non-zero (i.e. more to read)
          WHILE (tcount = tcount-1) GE 0 DO
                  BEGIN
-- reset OCB
                  CONTROL_OFF;
-- wait for ICB to set
                  WHILE CONTROL_IN EQ 0 DO 0;
-- set OCB
                  CONTROL_ON;
-- wait for ICB to reset
                  WHILE CONTROL_IN EQ -1 DO 0;
-- load next word
                  BUFFER(ptr = ptr + 1) = SENSIO(IN_BUF_LIO,INT);
          END; -- while loop
-- clear OCB, exit
          CONTROL_OFF;
          END; -- routine RECEIVE
```

Figure A6

AML code for the **RECEIVE** routine which receives data sent by the MINC DRV11 to the IBM RS1 robot system.

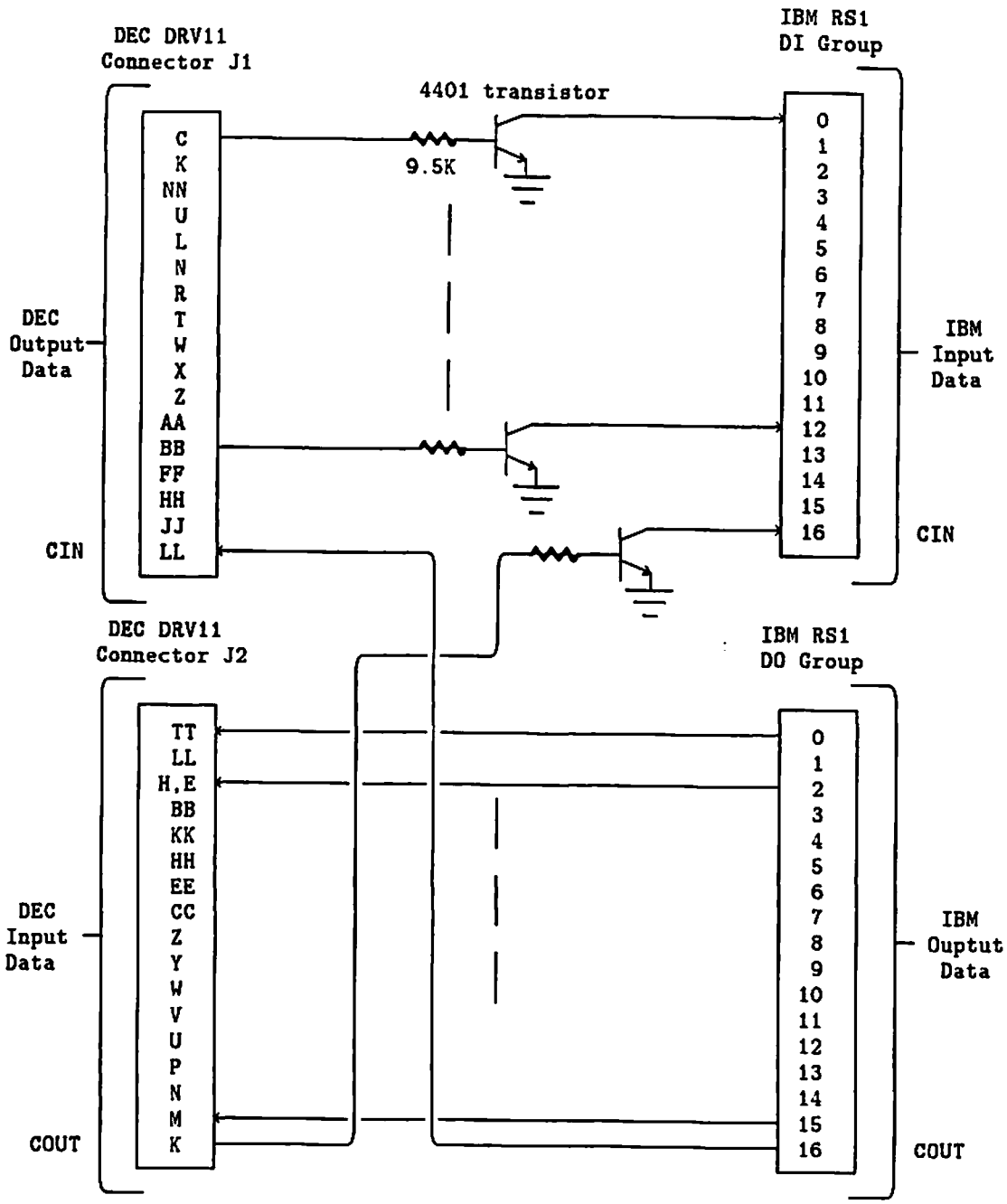
```
--
-- Special note: IBM uses 0-volt true logic ... the following
--                routines invert the control bits only

-- RS1 I/O port definitions
IN_BUF_LIO: NEW DEFIO(21,0,1,0,16);
OUT_BUF_LIO: NEW DEFIO(25,-1,1,0,16);
IN_CONTROL_LIO: NEW DEFIO(22,0,1,15,1);
OUT_CONTROL_LIO: NEW DEFIO(26,-1,1,15,1);

-- + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
-- "CONTROL_IN" returns the value of the input control bit
CONTROL_IN: SUBR;
                return(SENSIO(IN_CONTROL_LIO,INT));
        END; -- routine CONTROL_IN

-- + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
-- "CONTROL_OFF" turns the output control bit off (i.e. 0 volts)
CONTROL_OFF: SUBR;
                SENSIO(OUT_CONTROL_LIO,1);
        END; -- routine CONTROL_OFF

-- + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
-- "CONTROL_ON" turns the output control bit on (i.e. 5 volts)
CONTROL_ON: SUBR;
                SENSIO(OUT_CONTROL_LIO,0);
        END; -- routine CONTROL_ON

-- + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
-- This clears the outputs when the file is loaded
SENSIO(OUT_BUF_LIO,0);
CONTROL_OFF;
```

Figure A7

AML code for the primitive function definitions used by the routines which provide the IBM side of the communication interface, allowing the transfer of data between the IBM RS1 robot system and the MINC DRV11. Synchronization is provided via a control input bit and a control output bit which are implemented as the DI16 and DO16 bits, respectively.

Figure A8

Schematic of hardware Connections comprising the interface between the DEC MINC system and the IBM RS1 system. The control input bit, CIN, on the DEC side is the REQA bit of the Control Status Register and appears on connector J1 as pin LL while the control output bit, COUT, is the CSR0 bit of the CSR and appears as pin K on connector J2. The CIN bit on the IBM side is DI16 while the COUT bit is implemented as DO16.


DEC DRV11
Connector J1
4401 transistor
IBM RS1
DI Group
C
K
NN
U
L
N
R
T
W
X
Z
AA
BB
FF
HH
JJ
LL
9.5K
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
DEC
Output
Data
IBM
Input
Data
CIN
CIN
DEC DRV11
Connector J2
IBM RS1
DO Group
TT
LL
H,E
BB
KK
HH
EE
CC
Z
Y
W
V
U
P
N
M
K
DEC
Input
Data
IBM
Ouptut
Data
COUT
COUT

Appendix B

Sensor Design Extensions

B1 Introduction

The sensor presented in Chapters 2 and 3 of the dissertation is designed to transduce the component of the force orthogonal to the surface of the detector. This component of the human sense of touch is but a small part of the total somatosensory system. Other components of the system include the sensing of temperature, pain, thermal conductivity, vibration, shear, etc. This appendix explores a number of possible extensions to the original design.

The section B2 discusses an extension of the original design in a sensor designed to transduce shear forces while section B3 presents a whisker or "hair" sensor. The final section, B4, provides an in depth discussion of the Pacinian corpuscle mentioned briefly in Chapter 1 and presents some thoughts regarding how an electro-mechanical analog of this receptor could be used in a robot system.

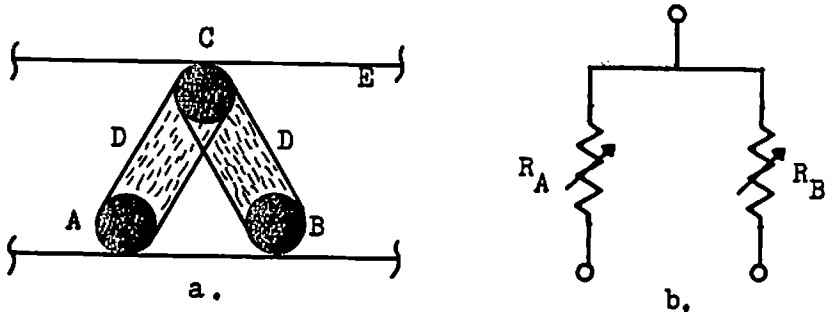


Figure B1

This shows a cross section and equivalent circuit for a sensor designed to transduce shear forces. B1a shows the components of the device which include two bottom conductors (a and b), a top conductor (c), and two carbon-filled rubber elements (d) contained in a non-conductive rubber supporting matrix (e). B2 shows the equivalent circuit which simply contains two variable resistors, Ra and Rb, representing the resistances through the two rubber elements (d) measured between the conductor pairs (a,c) and (b,c) respectively.

B2 A Shear Sensor

The first extension of this sensor involves using the basic ideas of that sensor to produce one which is capable of transducing the component of the force tangential to the sensor surface. Such information is extremely useful in understanding slip, for example.

As in the basic sensor, see Chapter 2, a carbon-doped elastomer is used to transduce the force on the detector into a resistance. Each element is composed of one top and two bottom conductors with the electrical path between the two being carbon-doped rubber columns. Figure B1 contains a cross-sectional view of one element in the sensor.

There are three simple ways in which the sensor can be deformed by forces in the x-y plane. In the first case, the sensor is deformed by a force normal to the pad, see Figure B2. In the resting state, identical columns will have resistances Ra, Rb and their ratio, Ra/Rb = 1. Applying a force normal to the surface would deform the two columns equally. The new resistances would be Ra' and Rb' and the ratio Ra'/Rb' would remain 1.

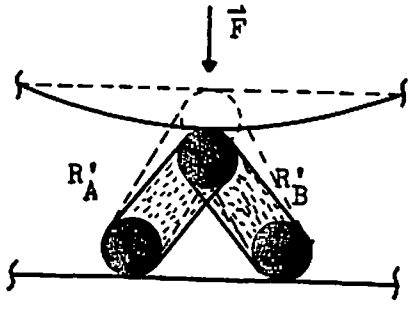


Figure B2

This shows the deformation of the sensor in Figure B1 due to a normal force. In the resting state, identical columns will have resistances Ra, Rb and their ratio, Ra/Rb = 1. Applying a force normal to the surface would deform the two columns equally. The new resistances would be Ra' and Rb' and the ratio Ra'/Rb' remains 1.

Figure B3 illustrates the deformations due to forces tangential to the upper surface. Figure B3a illustrates the case wherein the shear force is directed to the right. In this case, column B is compressed, resulting in Rb'>Rb, while column A is extended, making Ra'<Ra. The new ratio of the resistances is then Ra'/Rb'>1. The corresponding configuration with the tangential force in the opposite direction is given in Figure B3b. Here the ratio Ra'/Rb'<1.

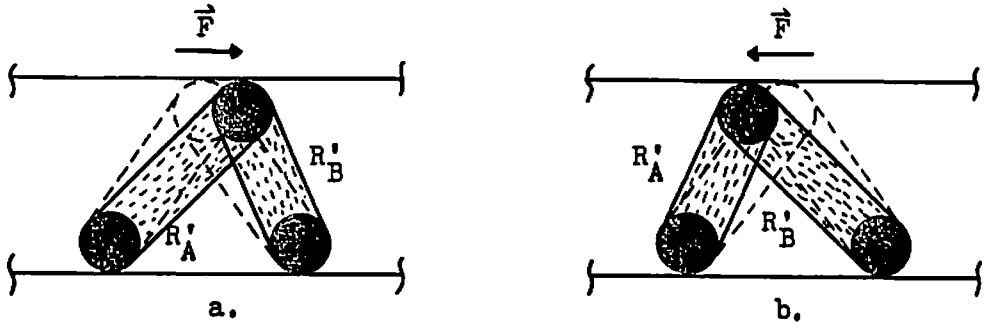


Figure B3

This figure illustrates the deformations due to forces tangential to the upper surface. B3a illustrates the case where in the shear force is directed to the right. In this case, column B is compressed, resulting in Rb'>Rb, while column A is extended, making Ra'<Ra. The new ratio of the resistances is then Ra'/Rb'>1. The corresponding configuration with the tangential force in the opposite direction is given in B3b with the resulting ratio: Ra'/Rb'<1.

Several factors enter into the functioning of this sensor. The lateral deformation of the top surface of the sensor due to contact with an object is dependent on the surface characteristics of both the sensor and the object, the force applied by the object normal to the surface, and the elasticity of the rubber materials composing the sensor. It appears that the most efficient method for understanding the characteristics of the sensor is to calibrate each one experimentally.

The above description focused on the concept. In practice, this sensor would use the same design as the basic sensor, e.g. the use of a diode in series with each column and the measurement actually being made of the voltage across a dropping resistor, see Figure B4.

In the full sensor, several of these individual sensors would be arranged in a matrix to provide an image of the tangential forces across the surface. A second matrix aligned to detect forces along the other dimension of the surface could be embedded in the first to provide the 2D information. (The component normal to the surface would be sensed by original sensor.)

B3 A Whisker Sensor

The second extension again involves using the basic concepts from the original sensor but in this case the intent is to produce a CONTACT proximity whisker sensor. Such a sensor will provide information about the presence of an object in the immediate vicinity of the gripper. The design of this sensor was motivated by the structure and function of hair/nerve ending systems in animals.

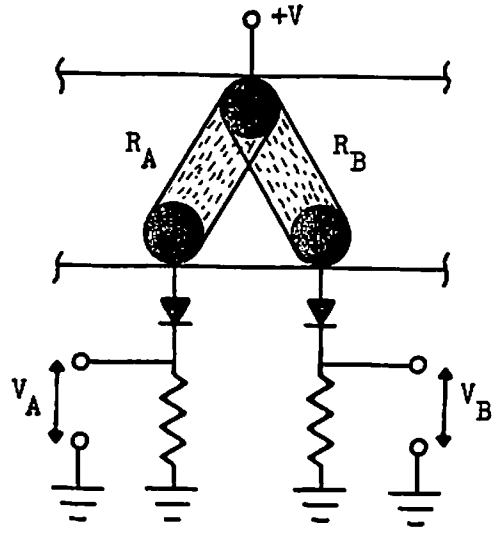


Figure B4

This is a schematic of a possible implementation employing the diode and dropping resistor approach used in the sensor discussed in Chapter 2.

The purpose of the sensor is to provide a means by which information can be obtained about an object close to, but not in direct contact with, the robot arm/gripper can be obtained. The sensor consists of a base with a protruding "hair" shaft or whisker. Deflection of the whisker from its resting position can be detected. A cross-sectional view of the sensor is given in Figure B5.

This is a modification of the shear sensor shown in Figure B1 and the analysis is identical. Shown in B1a are two variable resistors, Ra and Rb, composed of carbon-filled rubber. The bottom contacts a and b in Figure B1 are contained here as the outer contacts a and b; the inner contacts, c, are analogous here to the top contact in the shear design. A rigid "hair" fiber extends from the middle of the sensor. A side force on the fiber produces a torque which is resisted by the carbon-filled rubber. The shaft is located radially by the plate on the bottom of the sensor. The torque produces a partial compression and extension of the two resistors resulting in a change in the magnitudes of the resistances. Inset A in Figure B5 illustrates how the hair sensor can be used to provide the same information as the tangential sensor previously described. (The hair shaft is attached to a sheet, "Q" in the figure, which covers the base of the sensor. Movement of layer Q results in deflection of the hair.) Deflection of the hair shaft to the left in the figure will result in compression of the carbon-doped rubber resistor, Ra, resulting in a drop in the

resistance. By measuring this change in resistance as an increase in the voltage across a dropping resistor, shaft deflection information is obtained. Figure B6 illustrates the complete sensor and the equivalent circuit including the diodes and dropping resistors.

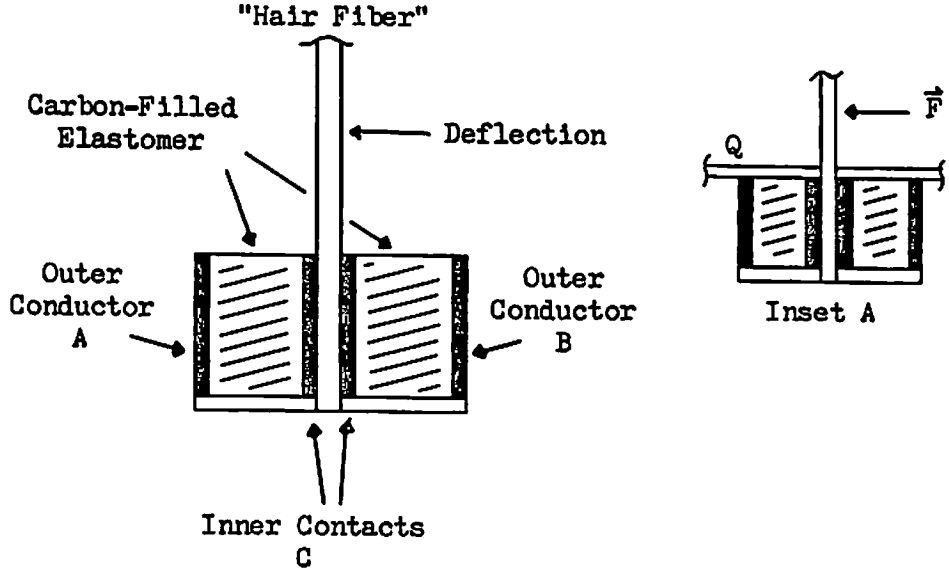


Figure B5

A cross-section of a possible whisker sensor is shown in the figure. It is a modification of the shear sensor shown in Figure B1. Shown in B1a are two variable resistors, Ra and Rb, composed of carbon-filled rubber. The bottom contacts a and b in Figure B1 are contained here as the outer contacts a and b; the inner contacts, c, are analogous here to the top contact in the shear design. A rigid "hair" fiber extends from the middle of the sensor. A side force on the fiber produces a torque which is resisted by the carbon-filled rubber. The shaft is located radially by the plate on the bottom of the sensor. The torque produces a partial compression and extension of the two resistors resulting in a change in the magnitudes of the resistances. Inset A shows the addition of a "skin", Q, over the sensor through which the fiber protrudes.

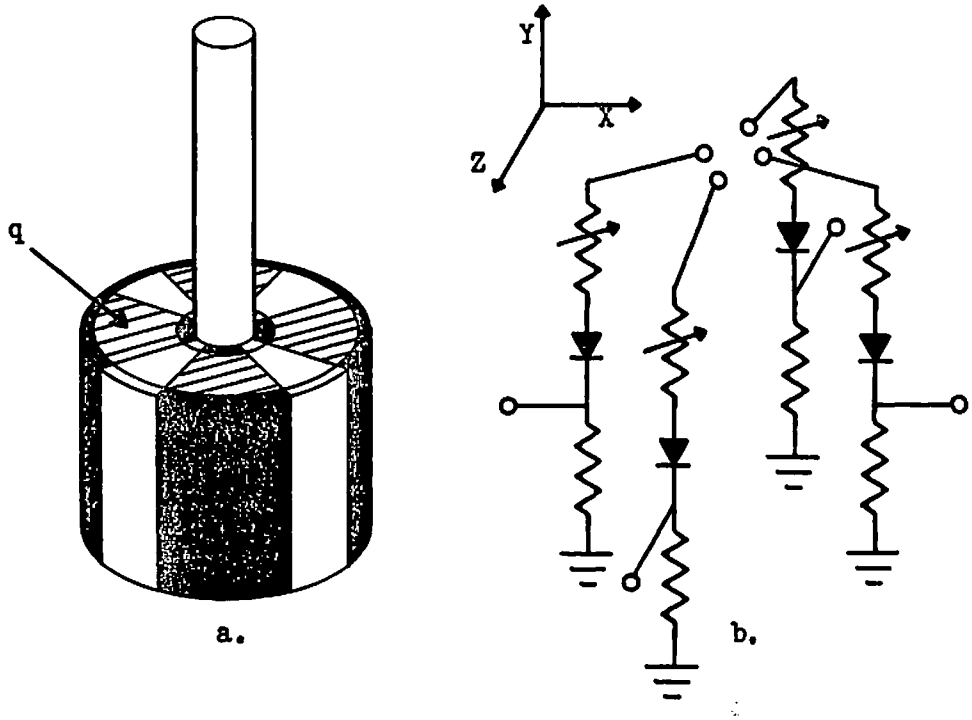


Figure B6

The diagram in B5 was a cross-section of the device and thus showed only two resistors. The drawing in B6a shows how such a device might appear in three dimensions. Four variable resistors, q, are shown with four side contacts, r. The resistors are insulated from one another by rubber filler. The side contacts are also electrically isolated. B6b shows the equivalent circuit for the device with the diodes and dropping resistors added.

B4 The Pacinian Corpuscle and Touch

B4.1 Introduction

An interesting approach to the design of robot sensory systems involves the analysis of the human somatosensory system. Applying the principles derived from such study to the design of processing architectures and software could lead to useful machine systems. In particular, what types of information from the periphery are provided to the Central Nervous System (CNS), and how is

the information processed while in route to the CNS? Chapter 1 contained a discussion of the afferent nerve responses to environmental stimuli and the receptors attributed to the nerves. Unfortunately, it is not the case that all of the receptors giving rise to particular nerve responses are known.

The topic of this section is one of the receptors whose response characteristics are known. This section is devoted to a detailed discussion of the Pacinian corpuscle which clearly appears to respond to vibrations. In addition, it presents a possible electronic equivalent sensor for use by robots and discussions of possible users of such a sensor. Finally, some thoughts on the use of the Pacinian corpuscle information by animals are presented.

B4.2 The Pacinian Corpuscle

Recall from the discussion in Section 2.2.2 that the skin of mammals may generally be considered to be one of two types: glabrous or hairy. The glabrous, or hairless, is found on the palms of the hands, bottom of the feet, lips, etc. In both cases the structure of the skin consists of an outer protective layer known as epidermis and an inner vascularized layer, the dermis, containing the majority of the tactile receptors. (Each of these layers is actually composed of several layers.) Pacinian corpuscles are found primarily in the hairy skin also in glabrous skin, deep regions such as connective tissue in the abdominal cavity [112], and, in a slightly modified form, in the muscles and joints [210]. In the skin they tend to reside near the base of the dermis - a reasonable distance from the environment.

B4.2.1 Structure

The Pacinian corpuscle appears as an encapsulated nerve ending and was first reported by Vater in 1741, then rediscovered by Picini (from whom the name was derived) in 1835. The corpuscle was originally described as having several lamellae separated by an unknown fluid and connected by ligaments. Much of our current understanding is due to a study by Cauna [42] of samples of Pacinian corpuscles taken from a group of human subjects whose ages ranged from newborn to 93 years. The corpuscle is composed of a multi-layer, ellipsoidal sheath containing numerous cells which surrounds a gelatinous core.

The sheath is divided into an outer section known as the "peripheral zone" [127] which surrounds the "inner core", with the differentiating feature being the thickness and spacing of the lamellae. (An excellent series of electron micrographs of encapsulated receptors is given in [43]. Although the Pacinian corpuscle is not explicitly considered, the treatment of general encapsulated receptors is very interesting.) The lamellae are typically less than 1 micrometer in thickness and composed of a single layer of cells. The capsules are typically 500-700 micrometers in length and 30 micrometers in diameter at birth and gradually increase in length to 3-4 mm. The capsules vary greatly even within an individual, yet they typically have rounded distal ends with the proximal ends tapering into the supplying nerve. The growth tends to assume the form of the addition of sheathing layers and retrograde growth along the axon. After

the age of 70 years, the size gradually decreases with an accompanying increase in the irregularity of shape.

The corpuscles are innervated by large myelinated fibers exhibiting conduction velocities in the range of 25-45 meters/sec [68] which provide uninterrupted pathways to the CNS. In fact, it appears that one of the functions of the mossy fiber system is to transmit Pacinian corpuscle activity directly to the cerebellar cortex [112]. (For a collection of papers giving a more complete discussion of the central pathways, the reader is referred to [39].) The unmyelinated end of the nerve is at the center of the gelatinous core. Innervation densities are known for the human hand [106] and are as follows: 21.4 corpuscles per sq. cm in the fingertip, 9.5 per sq. cm in the rest of the finger, and 9.3 in the palm of the hand. Fibers ending in such structures respond optimally to vibratory stimuli as will be detailed in the following section. Studies on the proportion of capsule mass to nerve mass have yielded figures indicating that the total volume of a Pacinian corpuscle in the adult human can reach 0.0001 cc yet 99.9% of that volume is accounted for by the capsule structure and not the nerve [127].

Loewenstein attempted to understand the mechanism involved in the transduction of mechanical stimuli to nerve activity, eventually concluding that the action potential developed at the first node of Ranvier and not within the capsule. He further deduced that the capsule itself played no part in the production of the generator potential leading to the spike. Loewenstein continued by investigating the response of extracted Pacinian corpuscles after having sections of the sheath removed. He found no apparent change in the stimulus-response characteristics even after complete removal of the peripheral zone and laceration of the inner core sheathing. Considering this and the fact that bare nerve endings respond to almost any physical stimulus, the purpose of the laminar sheath appears to be to act as both a filter and a protective cloak. The structure must preferentially attenuate transient stimuli depending upon their frequency.

B4.2.2 Function

As stated in Chapter 1, the response of nerves ending in these structures seems to indicate that they act as transducers for transient or vibratory stimuli. They are considered as belonging to the general class of fast adapting fibers since they exhibit virtually no response to static depression of the skin. Their response is typically in the range of 10-1000 Hz [186]. Sato found the minimal frequency eliciting a response was in the range of 10-20 Hz. The optimal response appeared between 150 and 200 Hz and the corpuscles failed to respond beginning at 500 Hz, depending upon the temperature. (At 24-27 degrees C, the upper limit was 500 Hz. This limit approached 1000 Hz as the temperature was increased to 34 degrees C.) Within this band, the output of the receptors seems to faithfully echo the stimulus with one or a small number of spikes per vibratory cycle [68]. They show no directionality as individual units, yet it should be mentioned that it is possible to derive directional information from fields of such

detectors (see below, Section 2.2.3). They have been shown to respond to indentations in the skin as small as 1 micrometer.

The typical psychophysical threshold vs. frequency curve has two distinct sections: a flat response section in the lower frequency range and a "U" shaped response curve at higher frequencies which is attributed to the Pacinian corpuscles. Von Bekesy in 1939 and, later, Verrillo [209] termed this a duplex theory of vibratory perception.

In the lower frequency ranges, up to 90 Hz, the threshold is independent of frequency. It is postulated that this section could be due to the response of the Meissner corpuscles [72] but no evidence is available to support this notion. The minimum threshold occurs at roughly 200 Hz with the threshold increasing with frequency thereafter. Several experiments have been constructed to test the duplicity theory which, simply stated, maintains that two mechanisms are involved in the perception of vibrotactile stimuli. In an interesting experiment, Gescheider attempted to separate the two systems by selectively adapting first one then the other [72]. (When the linear system is adapted, the threshold follows the expected curve for the Pacinian corpuscle. This appears to be very clear evidence that, in fact, the perception of vibratory stimuli is accomplished, in part at least, via a two component system.)

The next question to be asked concerns the size and shape of the receptive field of the corpuscles. Several studies [106], [114], [116], and [206] have concentrated on this question and the results have been consistent. In contrast to many other receptors, the fields of the Pacinian corpuscles are extremely large, sometimes encompassing the entire hand.

The receptive field is defined as the area over which any response could be elicited from the fiber. Over the interior of the field, the response shows a spot of maximal sensitivity near the center with a slowly increasing threshold as one moves towards the edge [107].

Given the frequency tracking of the response and the receptive field characteristics, it is clear that the magnitude of the stimulus is not encoded by the firing rate of the unit. The current belief is that magnitude is encoded by recruitment of additional receptors/nerves [68]. That is, as the stimulus increases in magnitude (increased force, not frequency), corpuscles in the immediate neighborhood begin to respond. Alternatively, the magnitude may not be encoded by these elements at all.

Other receptors, e.g., pressure, would also be responding to the vibratory stimulus, and their responses would vary relative to their response to strict pressure. Consider the analogy of a DC voltmeter being used to measure an AC input. The CNS could use Pacinian corpuscle output as an indication of the presence of a vibratory stimulus and derive the magnitude from other detectors' input combined with the Pacinian corpuscle input.

B4.2.3 Relationship to Robotics

In animals, touch or tactile sensation assumes many forms: pressure, temperature, vibratory, shear, etc. All of these submodalities are transduced by

a large number of receptors covering the entire surface of the organism. Research in the use of tactile data by robot manipulators has been limited mainly to gross force sensing in the wrist, and in some cases the fingers. In robots, simple binary touch sensors in the jaws of a manipulator were used by Paul and Bolles [32] to establish the presence or absence of a part. A robot hand with contact sensing on several sides was developed by Goto [84] to handle blocks placed on a table. In later work, Garrison and Wang [69] constructed a hand containing an array of contact sensors. In 1971, Okada and Tsuchuya experimented with the use of an artificial hand in the recognition of spatial patterns [154]. Hanafusa and Asada developed a hand in which each of three fingers contains a pneumatic sensor at the tip and described an algorithm allowing the hand to center itself over arbitrarily sized cylinders [88]. Recent sensor arrays have utilized conductive rubber as the physical transducer. A high spatial resolution conductive rubber contact sensing array has been developed at the Artificial Intelligence Laboratory at M.I.T. and has been successfully used to identify a bolt from its contact pattern. A hand constructed by Hill and Sword [97] employed analog force sensing arrays. Takeda constructed a hand with arrays of free-floating needles as sensors. The needles would conform to the shape of the surface being touched, thereby providing identification information [193]. The Laboratory for Automation and Systems Analysis at CNRS in Toulouse, France, has developed an "artificial skin" tactile sensor suitable for gross object recognition [34]. As indicated in the body of this work, tactile sensing in robots has been limited to force sensing in the major members, occasionally contact sensing in the fingers, and, in a small number of cases, arrays of force transducers in the fingertips. Efforts described here have extended the work in array sensing; however, in all cases the spatial resolution as well as the sensitivity is much less than that of the biological receptors. Usually only the grippers have any sensing capabilities and in no case is the entire surface of the robot represented in the tactile modality. As in machine vision, the computer controlling the robot receives the raw sensory information with no preprocessing. What is more important in light of the current discussion is that all of the forms of tactile sensation are static in nature. Even the work in Chapter 5 was based on a set of static images, and thus provided no direct information regarding transient phenomena. In many situations vibration information would be beneficial.

For a simple example, consider the fingerprint on the human hand. The print consists of a pattern of ridges and valleys. Some initial thoughts on the function of the ridges from their underlying structure have been presented by Couna [46]. He considered the cellular structure and concluded that the papillary ridges served the purpose of increasing the mechanical activity of the receptors via leverage. The comments here are directed at what one might call "global" level. The detailed anatomical structure of the skin is not considered but rather it is assumed to be composed of a homogeneous substance. A stimulus moving in a direction perpendicular to the ridges would produce a vibratory stimulus which could be transduced by the Pacinian corpuscles, see Figure B7.

This particular pattern contains roughly 22 ridges per cm. In order to

produce a vibratory stimulus of 250 Hz (minimum threshold for PC), a sharp stimulus moving along line "m" would have to travel with a velocity of approximately 10 cm/sec. While the receptive field of a single Pacinian corpuscle is small relative to this speed, a network of spatially separated Pacinian corpuscles could be configured to respond preferentially to the direction of the moving stimulus, see Figure B8.

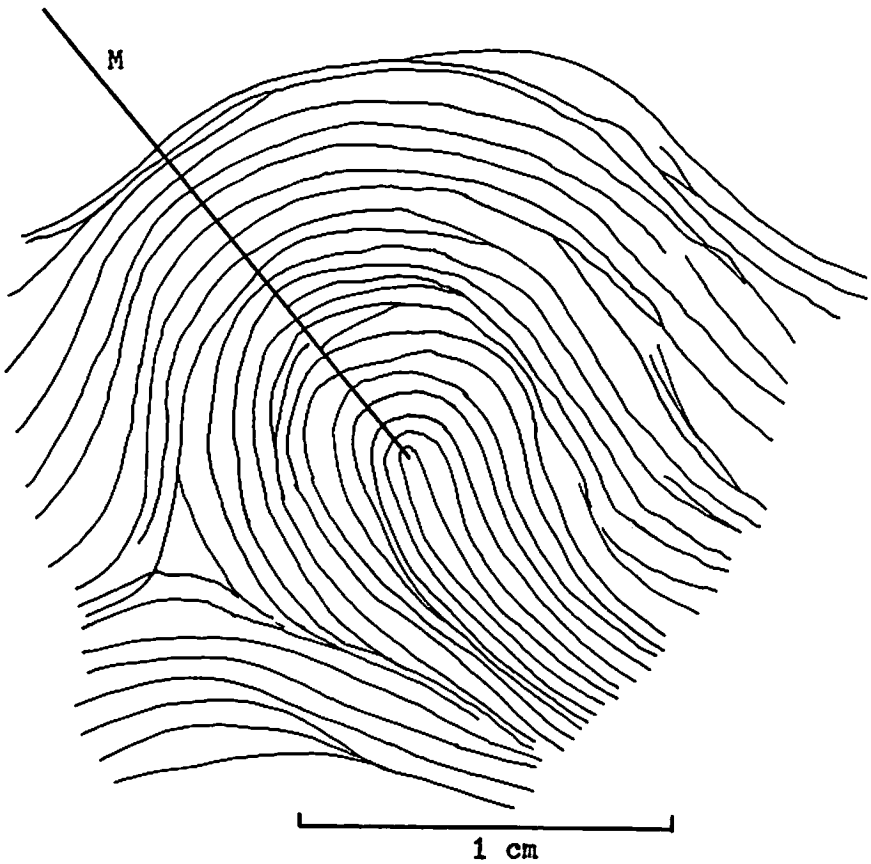


Figure B7

Typical human fingerprint

Figure B8 illustrates a typical conceptualization of such a network. Each receptor will respond regardless of the direction of movement of a stimulus when considered on an individual basis. The network consists of a set of receptors, ..., Rn-1, Rn, Rn+1, ..., which provides excitatory input to a bank of interneurons, In's, on a one-to-one basis. The output of each of the interneurons is felt by the two additional interneurons, Hn and Kn, which provide the lateral inhibition and excitation to actually give the network its directional sensitivity.

(For clarity in the figure, only Hn and Kn have been included. It should be realized that each interneuron Ij talks to such a pair.) Interneuron Hn has an inhibitory influence on interneuron In-1 and neuron Kn has an excitatory affect on In+1. The output of neuron Q will exhibit directional sensitivity.

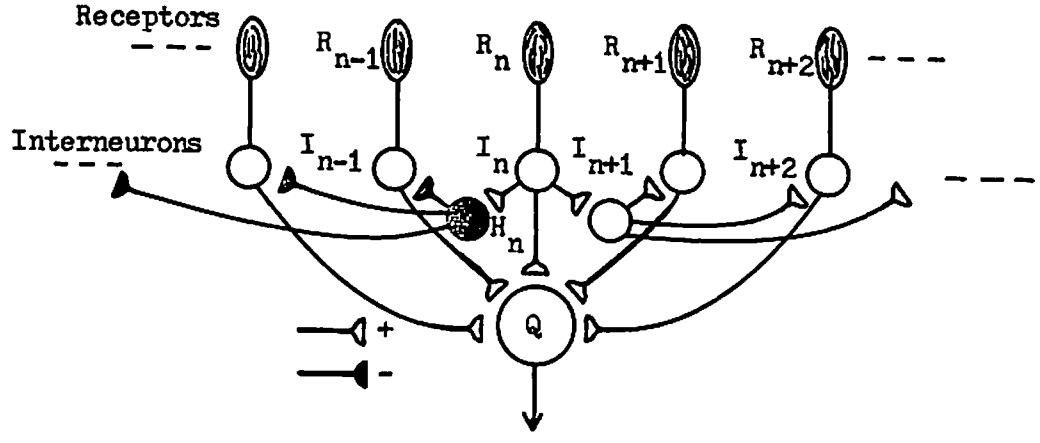


Figure B8

This figure shows a directionally sensitive network. Each receptor, Rn, will respond regardless of the direction of movement of a stimulus when considered on an individual basis. The network consists of a set of receptors, ..., Rn-1, Rn, Rn+1, ..., which provide excitatory input to a bank of interneurons, In's, on a one-to-one basis. The output of each of the interneurons is felt by the two additional interneurons, Hn and Kn, which provide the lateral inhibition and excitation to actually give the network its directional sensitivity. For clarity in the figure, only Hn and Kn have been included.

As a stimulus enters the receptive field of receptor n, Rn, it begins to respond. This in turn excites the neuron Q, indicating the presence of the stimulus to the CNS. Simultaneously, interneuron Hn is inhibiting In-1 and Kn exciting In+1. If the stimulus moves to the left, neuron In-1 will not fire due to the prior inhibition. Neuron Q's activity will decay to a resting level. If, however, the stimulus moves to the right, In+1 is sure to fire since it is also feeling excitation from In. In+1 then excites Q, inhibits In and excites In-2. The sensitivity of Q to the velocity of the stimulus can be varied by appropriate choices of the synaptic connection weights between interneurons, varying the transmission delay between the interneurons, and changing the range of lateral inhibition and excitation. Such circuits are seen in many biological systems. This

is provided as a demonstration of the possibility for use in the tactile system, not as the mechanism itself.

This type of sensing capability could be applied directly to robots. The fingertips of the gripper could be coated with a ribbed surface as in the fingerprint. A transducer responding to vibration could then be used to detect something moving over the surface of the gripper. A circuit emulating the Pacinian corpuscle is given in Figure B9.

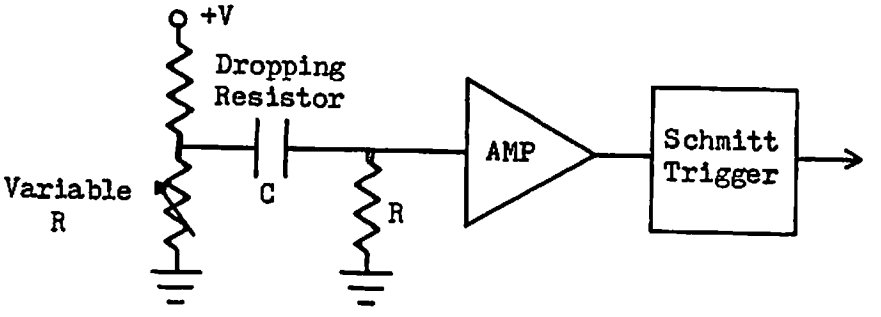


Figure B9

This contains a schematic for a possible Pacinian corpuscle emulator. The variable resistor is assumed to vary with pressure. See the text for an analysis of the circuit.

The variable resistor is assumed to vary with pressure. As the voltage across the resistor changes, due to a vibratory stimulus, the positive side of the capacitor will actually be "seeing" a signal with a small AC component superimposed on a DC signal. The purpose of the capacitor is to remove the DC offset. The amplifier then boosts the remaining AC component enough so that it can excite the Schmitt Trigger. The output of the trigger is a square wave which echoes the mechanical stimulus.

This appendix has presented several ideas for the extension of the sensor developed in Chapter 2 to sensors functioning in other submodalities of the tactile sense. Extended sensory capabilities will be required in order for future robots to function in a human world, and the addition of submodalities within touch other than normal force transduction is a beginning.

Appendix C

Psycho-physical Experiments

This work has addressed the questions surrounding robotic tactile sensing. An engineering approach was taken to the design of the tactile sensor and an artificial intelligence, or heuristic, approach used during the development of the processing techniques. While the physiology of the human tactile sensory system was touched upon briefly in Chapter 1, no explicit use was made of that information during the later work. This was due to the complexity of the human system and the current state of understanding of that system.

Robots will eventually be required to function in and among humans in human-tailored environments. It will be useful, then, for the machines to have the capability to manipulate the types of objects commonly handled by humans. The development of such capabilities will be aided by bringing to bear knowledge of the human nervous system at one level and an understanding of the higher level, cognitive aspects of human tactile perception. This appendix proposes some psycho-physical experimental work aimed toward correlating the cognitive terms "edge", "face", "point", etc. with the underlying environmental stimuli and toward understanding the tactile investigative behavior of humans. Two sets of experiments are outlined below which may provide some insight into these issues. The results could be relevant to the development of a robotic tactile perceptual system.

In the related research area of machine vision, one of the basic problems involves relating cognitive terms such as "edge" to the intensity data. The difficulty arises due to both the semantic complexity of the term and the inherent two-dimensional nature of the data from the three-dimensional world. In the tactile domain, at least, the latter of these problems is effectively eliminated, a tactile sensor provides three-dimensional information about three-dimensional objects. The question then becomes one of correlating the sensor data with the cognitive concepts. This first set of experiments is designed expressly for this purpose. It is posited that objects can effectively be described by the set of elements, e.g. edges, faces, points, etc., of which they are composed, along with their relationships. It seems unlikely that any of these terms have absolute definitions but more likely that the definition at any given time is dependent to some extent on the proximity and "strength" of nearby features. The following five experiments investigate the correlation of the concepts to the sensor data, both in isolation and in respect to one another.

P1: Tactile Edges

The first experiment in this set is designed to define the concept of a "tactile edge" in terms of the sensor data. A blindfolded human subject is presented a set of tactile stimuli consisting of two planar faces which share an edge. The subject is allowed to investigate the junction of the two surfaces with the index finger only, thus eliminating kinesthetic and surface shape information

from the other fingers. The parameters defining the junction between the faces are varied and the subject is asked to indicate how he/she would characterize it. For instance, the subject could be asked to choose one descriptive term from a set such as: edge, rounded corner, curved surface, flat surface, etc.

The stimuli would all be constructed from the same materials in order to remove the affects of surface texture variations, temperature variations, etc. The parameters to be varied include the radius of curvature of the junction and the angle between the faces. View each stimulus as being constructed from a solid block of material with the top face of the block machined into two flat surfaces joined together along a common edge which runs along the center of the block as shown in Figure C1.

P2: Tactile Points

This experiment is similar to the previous one with the exception that the concept to be studied is that of a "point." The stimuli here consist of a set of cones. The radius of curvature of the apex of the cone and the height of the cone are varied as the experimental parameters. A cross section through such a cone appears exactly as the cross section through the block stimulus used for the previous experiment. The radius of curvature of the apex of the cone corresponds to the radius of curvature, r, between the top surfaces of the block stimulus and the height of the cone to the angle, t, between the surfaces. These stimuli can be visualized as the solids generated by rotating the cross sections in Figure C2 about their axes of symmetry. The set of descriptive terms would be altered to include terms such as point, spherical surface, etc.

In the ideal case the joint between the top surfaces, A and B, appears as a line. The variable in this case is the angle, t, between the planes containing the surfaces. This angle can vary from 180 degrees, in which case the surfaces are coplanar and no edge is present, to theoretically, 0 degrees where the edge is a perfect razor blade. This range allows all convex edges to be generated. An interesting extension would involve allowing t to assume the range [180,360] degrees which defines all possible concave edges.

The second variable in the stimuli is the radius of curvature of the intersection of the two surfaces. The physical implementation of this is easily seen in the inset in Figure C1. Ideally, r could range from 0 to infinity, measured in any convenient unit of length. The first case represents the intersection of two planes and provides a discontinuity in the slope from one surface to the next. As the radius is made non-zero, the change in slope from one surface to the other becomes continuous; as the radius increases, the change in slope from one surface to the next becomes more and more gentle, i.e. the magnitude of the derivative of the slope decreases.

A set of stimuli would be constructed which covered a "reasonable" range of both parameters. Figure C2 contains 24 possible cross sections of the stimulus in Figure C1 taken along plane Q in the figure. The angle, t, between the surfaces is increased in steps of 15 degrees from one row to the next beginning with a value of 15 degrees. The radius of curvature is varied across the columns from an initial value of 0mm to a final value of 10mm in steps of 15 degrees.

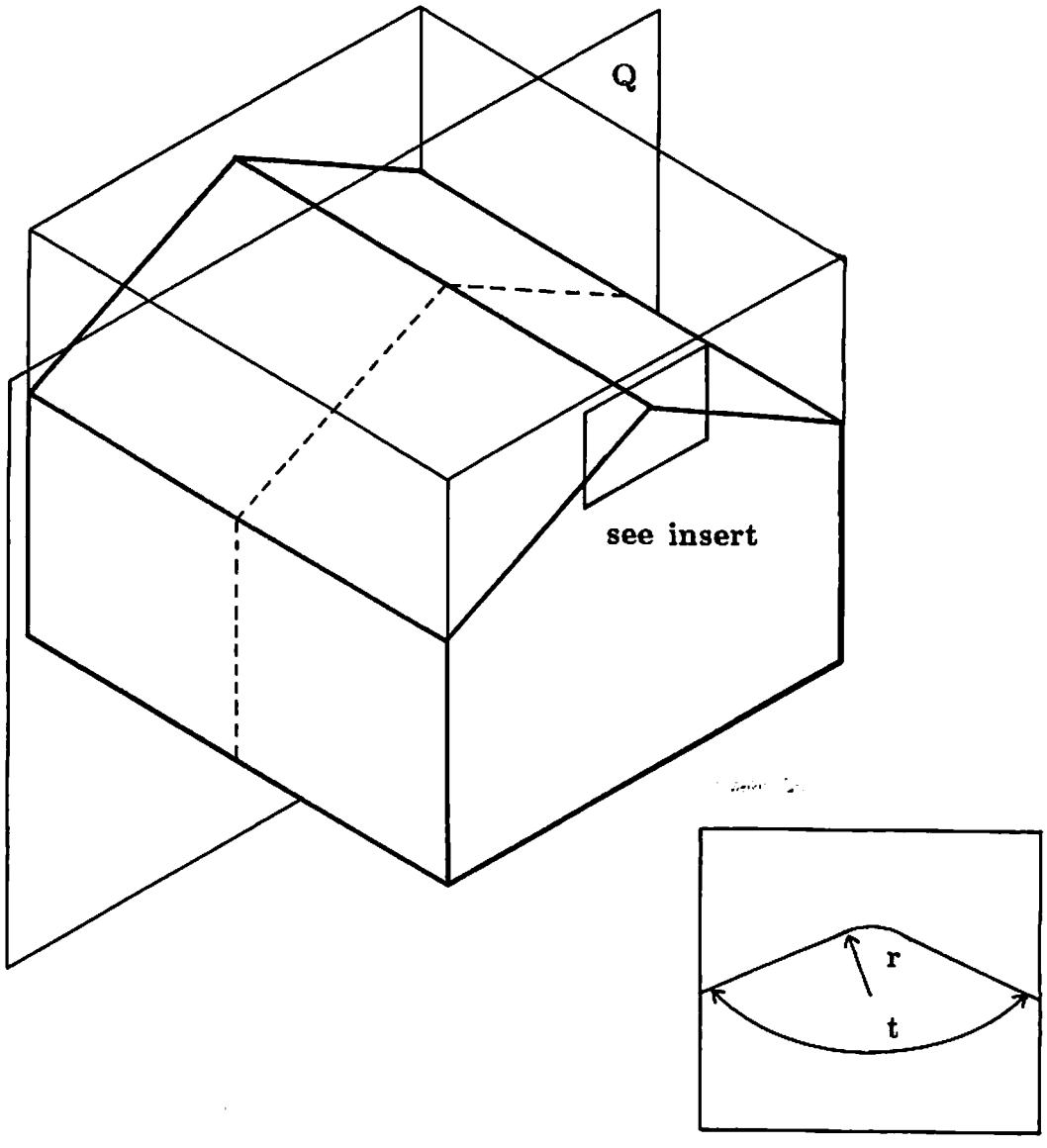


Figure C1

Generic stimulus used in experiment P1. The actual stimulus consists of the intersection of the two surfaces, A and B, forming the top of the block. This is characterized by the angle between the surfaces, t, and the radius of curvature, r, of the intersection.

Figure C2

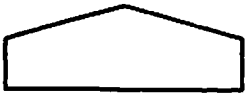
Set of 24 possible cross sections of the stimulus in Figure C1 taken along plane Q in the figure. The angle, t, is varied from 15 to 90 degrees in steps of 15 degrees along the rows. The radius of curvature, r, is varied across the columns from an initial value of 0mm to a final value of 10mm in steps of 15 degrees.

t=165 degrees
r=0mm

t=165, r=2

t=165, r=5

t=165, r=10

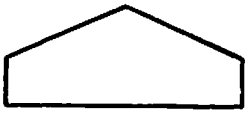


t=150, r=0

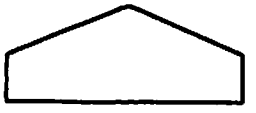
t=150, r=2

t=150, r=5

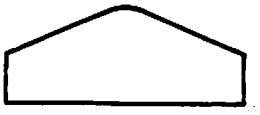
t=150, r=10



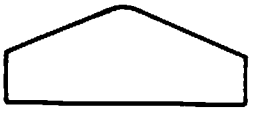
t=135, r=0



t=135, r=2



t=135, r=5



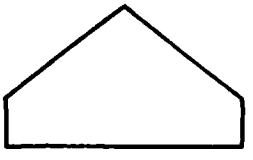
t=135, r=10

t=120, r=0

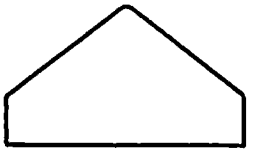
t=120, r=2

t=120, r=5

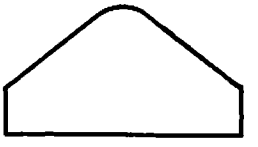
t=120, r=10



t=105, r=0

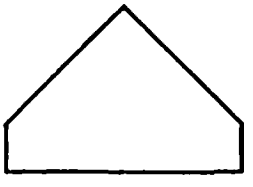


t=105, r=2

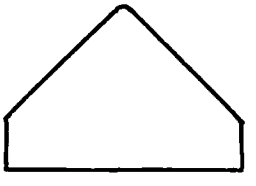


t=105, r=5

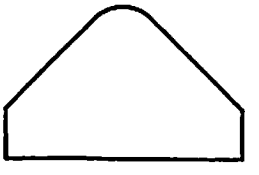
t=105, r=10



t=90, r=0



t=90, r=2



t=90, r=5

t=90, r=10

P3: Tactile Faces and Edges

The third experiment is intended to investigate the relationship between edges and faces. The intent is to understand the relationship between the size (width) of a face and the perception of the object in terms of edges or faces. The stimuli are right polygonal cylinders and all have minimum enclosing cylinders of the same radius. These stimuli can be viewed as extensions of those in the first experiment and thus there are two parameters of interest. The first is the angle between adjacent faces. Since the radii of the various stimuli are constant, this translates to increasing the number of faces. The second parameter is the radius of curvature at the intersections of the faces. The shape of the junctions between the faces will be chosen according to the results of the first experiment. That is, the radius of curvature should be chosen to be that which elicited primarily responses of "edge."

The subjects are asked to describe the shape of the stimulus cylinders in terms of edges and faces. It is expected that as the width of the faces and the "sharpness" of the edges vary, the descriptions will shift from being predominantly "a set of edges" to "a set of faces" to, finally, a "smooth cylinder". The points of the shifts relative to the parameter settings are the features of interest.

P4: Tactile Faces and Points

As above, the fourth experiment investigates the relationship of one concept to another. In this case the stimuli are regular polygons, e.g. a dodecahedron. The edges between adjacent faces are rounded sufficiently so that they do not enter into the subject's descriptions of the objects. Think of the edges as being smoothed to the point, as determined from experiment P1, that the predominant characterization is in terms of the faces. The "sharpness" of the vertices will be varied (see experiment P2) as will the size of the faces. The feature of interest is the point where the descriptions shift from having an emphasis on the points to placing the emphasis on the faces. At one extreme, the objects feel like pin cushions while at the other they feel like smooth balls.

P5: Tactile Faces and Edges

The final experiment in this set is a repeat of experiment P4 above with the edges designed to factor into the descriptions. The three parameters of vertex "pointedness," face-to-face junction "edginess," and face size are varied in order to understand their relationships to one another.

The second set of experiments is designed to investigate exploratory behavior in restricted environments. The interest here is in what Gibson has termed "Active Touch" [76]; that is, the conscious exploration of space. A series of behavioral experiments to be carried out in environments of increasing complexity is discussed below. It is hoped that an understanding of behavior in the constrained experiments will aid in the understanding of the later paradigms.

P6: Two-Dimensional Haptic Search

The first experiment is designed to investigate tactile exploratory behavior. In particular, the initial behavioral paradigm involves the exploratory behavior of humans in two dimensions. The subjects are asked to locate a hole in a flat surface with the tip of the index finger without the use of visual input. The positions of the index finger during the exploratory movements are tracked. This could be accomplished through filming of the activity and then digitizing the position of the center of the index fingernail during each frame.

The data could then be analyzed in order to determine the algorithm underlying the search pattern. Similar experiments could be undertaken to determine the form of the exploratory behavior exhibited during the location and tracking of the edge of a plate.

P7: Three-Dimensional Haptic Search

A direct extension to experiment P6 involves studying the exploratory behavior of humans in three dimensions. The subjects are presented with an object and asked to describe its shape in as much detail as possible. The movement of the index finger during the tactile investigation could be recorded on film for later analysis.

The question here is to define the salient features of the data which are extracted by the human, i.e. does the human track the edges of the object or are the faces the important structures. The results from this experiment would be twofold. First, an indication of the types of information important to human tactile exploration would be known. This could lead to the formation of hypotheses concerning the data structures into which the sensory information is integrated. Secondly, it is possible to develop algorithms for haptic investigation in three dimensions based on the observed behavior. The experiments outlined above are constrained to the extent that only the exploratory behavior is analyzed.

These two sets of experiments could provide some insight into the relationship between the physical structure stimulating a tactile sensory surface and the perception derived from that experience and search strategies employed during haptic search using only one finger. However, several points must be kept in mind regarding these experiments. Since the subjects used in these experiments are humans, many uncontrollable variables enter into the formation of a response to an experimenter's question. It is not clear that the experiments can be designed in such a way as to eliminate the influence of these uncontrollable variables and yet still provide meaningful information. In addition, the distinctions between the concepts such as "edge" and "point" may not be clear. One would expect that clear answers would be given at each end of the spectra in the first two experiments. The problem may arise in the middle, between the two extremes. There may be no clear-cut answers to the questions asked by the experiments.