A FAST MULTIPLE-TOUCH-SENSITIVE INPUT DEVICE

by

SEONKYOO LEE

A Thesis Submitted in Conformity with
the Requirements for the Degree of
Master of Applied Science
in the
Department of Electrical Engineering
University of Toronto

October 1984

ABSTRACT

Touch sensitive input devices in various forms have appeared in increasing numbers in the market place. Available devices, however, are deficient in the number of parameters that can be obtained from a human gesture. Most provide only the location of a single finger tip. But there are more parameters of interest, the pressure of a single touch or the existence of multiple touches for example.

This thesis describes the design and implementation of a fast-scanning multiple-touch-sensitive input device. This device utilizes a capacitance sensing technique in conjunction with a binary scanning algorithm for both flexibility and speed.

The resolution of the sensor matrix provided is variable to allow a tradeoff between resolution and speed. Even though the apparent maximum hardware resolution is fixed to be 64 by 32, the resolution can be further increased through the application of various interpolation schemes.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Professor K.C.Smith for the extent of his support and encouragement and to Professor William Buxton for his many useful ideas on the project.

I owe special thanks to my brother Seonjo Lee who helped me in building the touch tablet and to my wife Cheehyun Lee who has encouraged me through the last two years.

Grateful acknowledgement is given to the National Engineering and Science Research Council for scholarship support during the course of my study and for the Operating Grant support of my advisors, professor William Buxton and K.C.Smith.

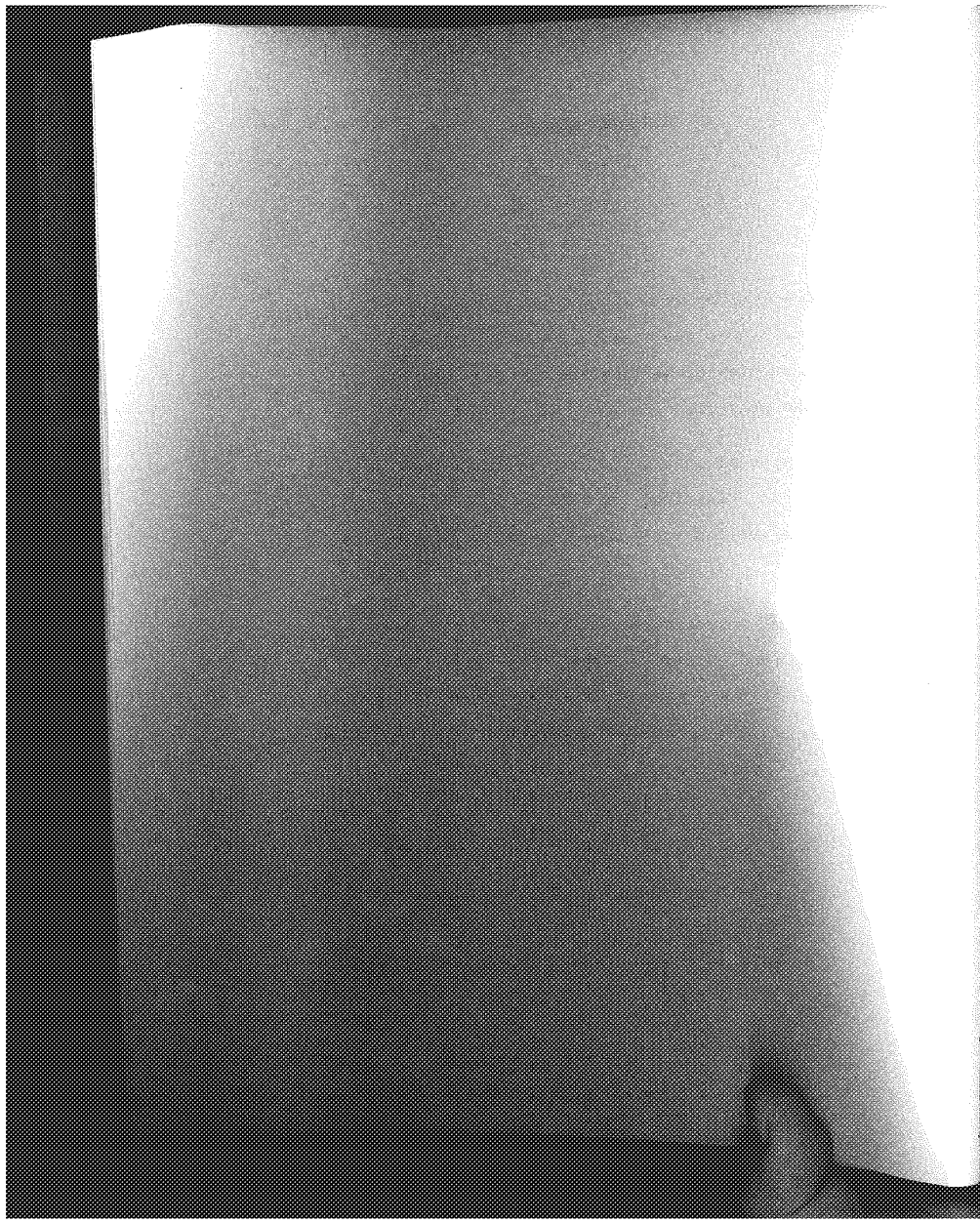


TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The rapid advancement of computer technology today has opened a variety of computer applications such as music, graphics, medical diagnosis, education, and text processing. Correspondingly, input devices to computer systems have grown quite numerous. A piano keyboard transducer, for example, is one such input device for music applications. Positioning devices are seen to be essential to graphics applications; image transducers are required for pattern recognition in medical diagnosis; touch screens are appropriate for the education of young children; while the Qwerty keyboard remains the usual standard for text processing. Since no single one of these specialized input devices fully utilizes the computer in some particular application, a combination of several different devices is seen to be useful and necessary.

A particular input mechanism of interest is finger motion. Input devices utilizing finger gestures are still numerous; to name a few, there are the Qwerty keyboard, the

Piano keyboard, various positioning devices such as the joystick, the light pen, the touch screen, and the graphics tablet. However, each of these devices is more or less specialized to a particular application as indicated above.

It would be desirable to have one or two devices serve all functions. However, a danger remains that such a device would not really suit all applications; there is the possibility that while it serves all, it pleases none. [ref 1.1] The solution of this difficulty seems to lie in a combination of the adaptability and flexibility of the device. Here adaptability refers to the reconfigurability of the device to a particular application, whereas flexibility denotes the ease with which a new application may be adapted.

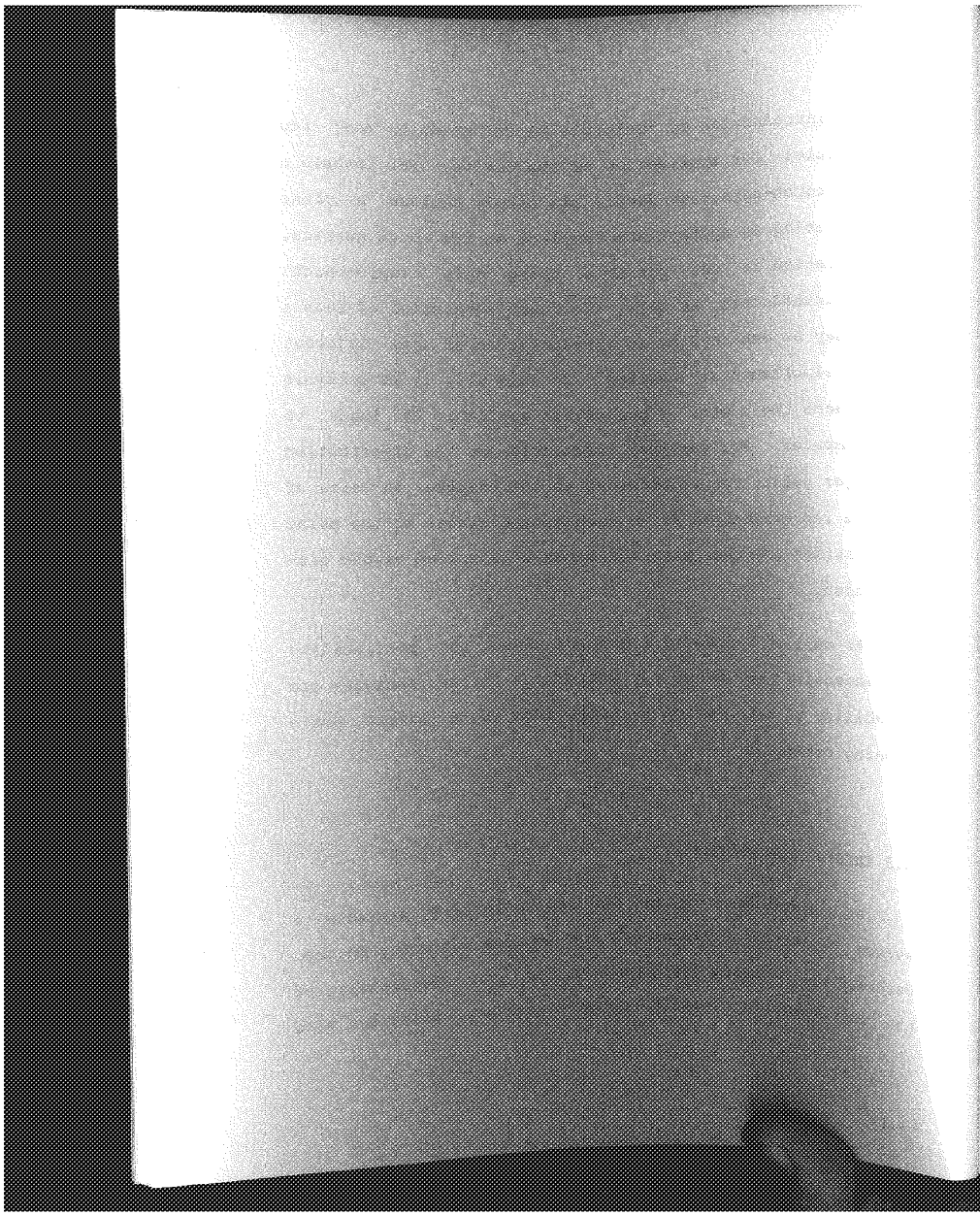
In an effort to find a general-purpose input device that is well suited for a wide range of tasks, many input devices and their transducers have been examined in this work. The result has been the development of a fast-scanning multiple-touch-sensitive input device. This device generates x, y and z coordinates for a particular touched position of a tablet surface utilizing a capacitance measurement technique. Unlike the previously developed capacitance touch tablet by L. Sasaki and G. Fedorkow at the University of Toronto[refl.2], it can detect simultaneous multiple finger touches without ambiguity.

The introduction of "multiple sense" opens up many new possibilities for application of the device. For example a multiple-touch-sensitive device can simply emulate a piano keyboard using an appropriate template on its touch surface. Such emulation is not possible using only single-touch-sensing transducers. As well, the touch resolution of such a tablet may be enhanced by an interpolation of many adjacent points simultaneously touched. In this way, it is possible to increase the number of parameters available as input to the computer. For example, information on the distribution of finger points can be provided to the computer in terms of a line representation drawn through the center of the point distribution together with a measure of a spread of the distribution.

As well of course in a general sense, the availability of intensity at each x,y coordinate further enhances the capability of this device in comparison to a simple positioning device.

1.2 THESIS ORGANIZATION

As a preliminary design process, chapter 2 presents a survey of touch transducers and devices currently in use. Hardware and software requirements for the development of the fast-scanning multiple-touch-sensitive tablet are also



presented. Hardware for the sensors and the corresponding interface circuits are described in chapter 3. Software and algorithms for the fast scanning technique are presented in chapter 4. Chapter 5 integrates the hardware and software together as a system for performance testing. As well, in this chapter, the sensors are characterized in terms of their performance. Finally the last chapter suggests future research and directions for the enhancement of the device in terms of its manufacturability and performance.

CHAPTER 2

DESIGN PROCESS

2.1 INTRODUCTION

A fruitful approach to developing a flexible multi-purpose input device is seen to lie in determining the amount of information that can be simply and conveniently provided through a single device. This is contrast to seeking ways of combining a number of existing devices.

The information that the user provides could be by means of physical gesture. To the user this implies repeatability such that the same gesture results in the same meaning, and dexterity in order that a variety of parameters can be produced with minimum sequence of expressions. The human hand can provide both required characteristics.

In particular, a simple natural communication technique is "TOUCH". Touch itself in general contains much more information than simply physical location. It could convey additional information such as, for example, the pressure or speed of touch of one or more moving fingers.

This chapter outlines the preliminary design processes leading to the development of a fast multiple-touch-

sensitive input device(FMTSID).

Section 2.2 presents five existing touch-sensitive devices with emphasis on their sensing mechanisms; section 2.3 discusses the shortcomings of the existing tablets; section 2.4 defines what must be done to the hardware and software to achieve FMTSID; and finally, section 2.5 concludes this chapter.

2.2 EXISTING TOUCH-SENSITIVE INPUT DEVICES

This section outlines existing touch tablet or screen sensing mechanisms emphasizing those using resistive, infrared, ultrasound, capacitive, and video techniques.

Resistive:

There are several approaches to the use of resistive sensors for pointing location on a tablet. Two specific and distinctive approaches are examined, namely those of the Elographics data tablet and a high resolution imaging touch sensor developed at M.I.T.

The Elographics data tablet [ref 2.1] uses a uniform resistive substrate with resistors attached to the edges in order to compensate non uniformity of the mapping between resistance and position. When the tablet is touched, pressure from the finger tip causes a deformable coversheet with

a conductive layer to contact the resistive substrate. A low voltage is applied alternatively along orthogonal axes to the edges so that each axis is exclusively selected to measure the voltage at the touch point which is a function of the distance from the selected edge. Figure 2.0 represents the functional models and construction of the Elographic tablet.

The M.I.T high resolution imaging touch sensor[ref 2.2] consists of a flexible printed circuit board, a sheet of unidirectionally uniform conductive silicone rubber and a separator to pull the conducting layer aprrt when pressure is released. The sensor arrays and electrical model of one row are shown in Fig. 2.1 and 2.2. The image of the touch is derived from the measurements of the impedances of all columns and rows in the resistor matrix. The individual impedance in a column [Fig 2.2] R[n], is obtained from successive measurements of the two port parameters of the subnetworks. The relevant equations are as follows:

$$Z[0] = 0$$

$$G[0] = 1$$

$$R[n] = \frac{V - I[mn] * G[n-1] * (Z[n-1] + Rl)}{I[n]}$$

$$Z[n] = R[n] * \frac{Z[n-1] + Rl}{Z[n-1] + R[n] + Rl}$$

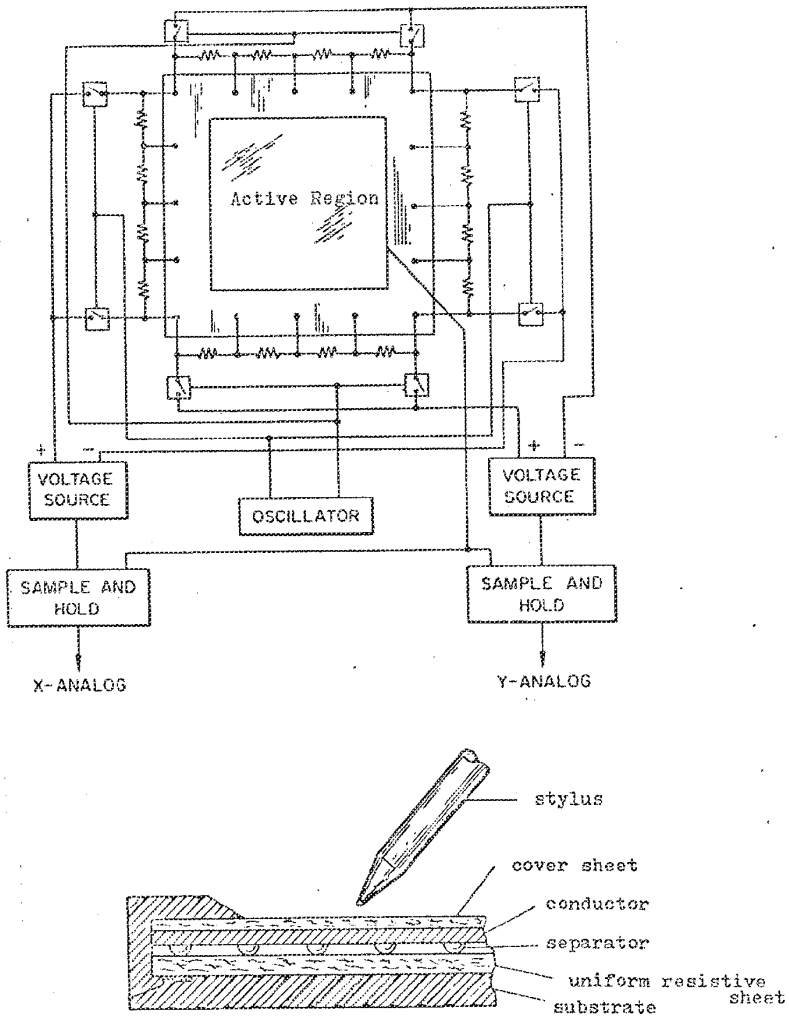


Fig. 2.0 Block Diagram and the Cross-section of the Elographics Data Tablet

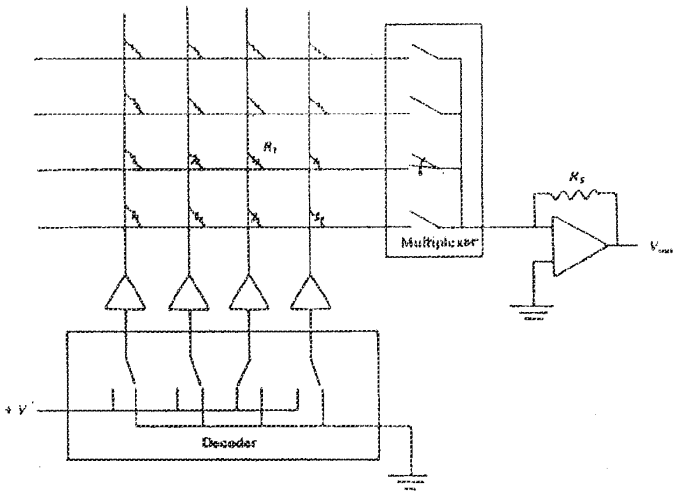


Fig. 2.1 Electrical Model of M.I.T. Imaging Touch Sensor

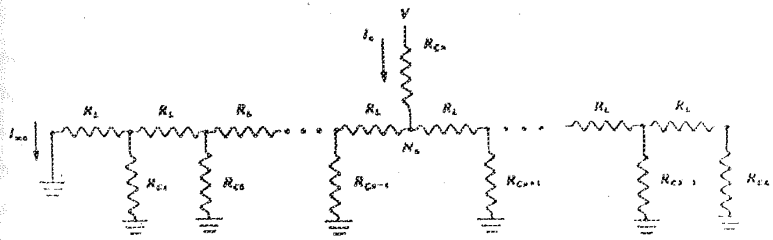


Fig. 2.2 Electrical Model of one Row of M.I.T Imaging Touch Sensor

Here V is the applied voltage, I[n] and I[mn] the measured currents, Z[n] the input impedance, G[n] the voltage transfer ratio of the network to the left of R[n], And Ri is the linear resistance of the unidirectional uniform resistive rubber sheet.

From these equations, the value of R[n] is obtained only when R[n-1] is known.

Infrared:

Infrared(IR) is one of the oldest techniques used in current sensing systems. A representative example in current use is that in the HP personal computer system.[ref 2.3] A number of the IR detectors and equal number of IR emitting diodes are mounted opposite to each other in horizontal and vertical directions such that the presence of a finger tip can be detected by the absence of the IR light at the horizontal and the vertical detectors.

Ultrasound:

The ultrasound sensor technique has been used in various areas. One of the applications in a touch sensitive input device employs piezo electric transducers mounted on horizontal and vertical edges of transparent glass screen. Here, the acoustic wave (Rayleigh wave) generated by the transducer travels along the free boundary of the glass, much like the ripple on a pond. A reflected wave or echo is

initiated by a touch on the surface. The time intervals between the arrival of the echo and the beginning of the transmission of the source signal, measured in two directions as shown Fig. 2.3 give the information on the position of the object. The touch sensor shown Fig. 2.3, developed by T.S.D Limited [ref 2.4], actually provides outputs of x+y and 2y enabling the derivation of x and y where x and y are the distance horizontally from a vertical edge and vertically from a horizontal edge respectively.

Video Technique:

The video approach to the touch input device uses a T.V camera to scan a translucent plate on which the finger tip is placed. The signal from the TV camera is then processed to obtain one bit of information per pixel. Using a programmable threshold voltage, one bit per pixel is obtained to determine the shape of a 2-D projection of an object on the plate. The data resulting from this process are stored in a memory to which access by a dedicated processor is allowed. This processor implements further processes such as determining non-zero locations in the memory (that is, the position of the object) and identifying shapes (grouping the pixels for an object). Nimish Metha presented such a system [ref 2.5] whose basic configuration is shown in Fig. 2.4.

Capacitive:

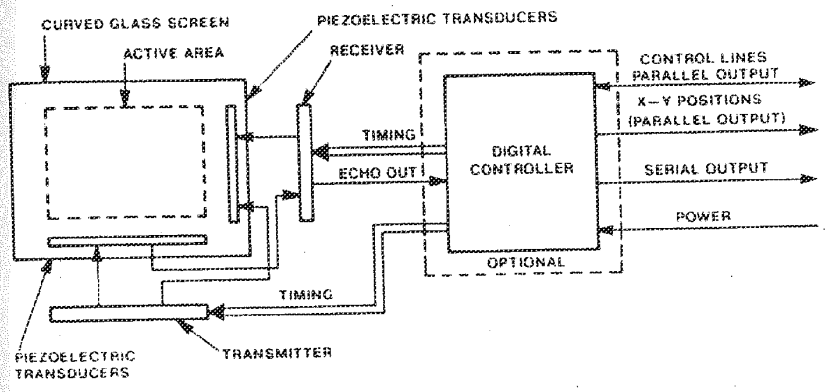


Fig. 2.3 Block Diagram of Touch Screen Digitizer

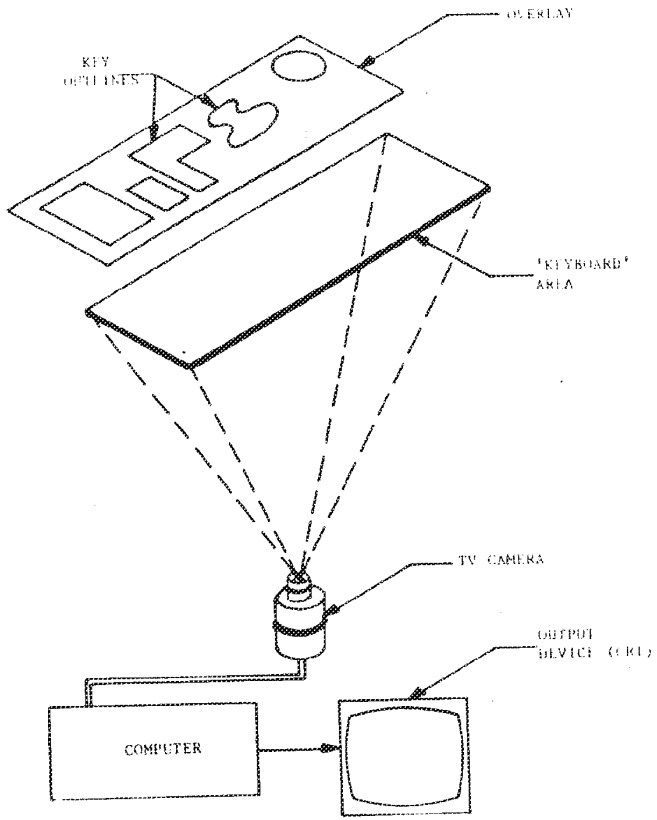


Fig. 2.4 Video System for Touch Tablet

Capacitive sensors are used in various applications such as single touch switches and touch tablets. Two kinds of touch tablet using capacitive sensors will be examined here. The one developed by TASA [ref 2.6] shown in Fig 2.5 measures the capacitance between the plate and the area covered by the touch. This measured datum is then compared with a previous reading stored in a shift register. In order to reduce the size of the shift register, a tablet is divided into many sub-regions in which the position of the touch is uniquely located. One such sub-region is much larger than the maximum size of a single touch so as to avoid an overflow. TASA utilizes the sensor to detect relative movements of a finger rather than its absolute position.

Another capacitance tablet developed by Sasaki, Fedorkow and others at the University of Toronto [ref 2.8] uses sensors for the rows and columns which are interleaved as shown in Fig 2.6. It uses analog multiplexors to select a row or a column sensor. In order to find the capacitance of a row or a column, it counts the time to charge up the capacitive sensor. Because the capacitance of the sensors on the tablet without a touch is not constant, and the capacitance change produced by a touch is relatively small compared to the capacitance of the surroundings, the system uses a measure of the initial capacitances of the sensors (without touch) whose values are stored in the memory for

reference. The touch point is determined by finding a set of the maximum difference values (current value less the reference value) for the rows and columns.

2.3 SHORTCOMINGS OF EXISTING DEVICES

The scanning properties of the devices described in the previous section can be distinguished depending on their position, pressure and multiple touch sensing capabilities. All the devices or transducers referenced are capable of locating the position of a touch, with a resolution which is characteristic of each device. Only capacitive sensors and the MIT resistive sensor provide pressure of touch whereas the video system and the MIT resistive sensor give a multiple touch capability.

Projective sensors, that is, all of the sensors introduced in the previous section except the video system and the MIT high resolution resistive sensor, cannot detect multiple touches without ambiguity since the detection of touch by two horizontal and two vertical sensors can be interpreted in seven possible ways as shown in Fig. 2.7.

In general, the missing data may be retrieved by introducing more axes, as is done in a tomographic imaging system. Using this scheme, if the upper limit of the number of touch points is two, the introduction of one more axis

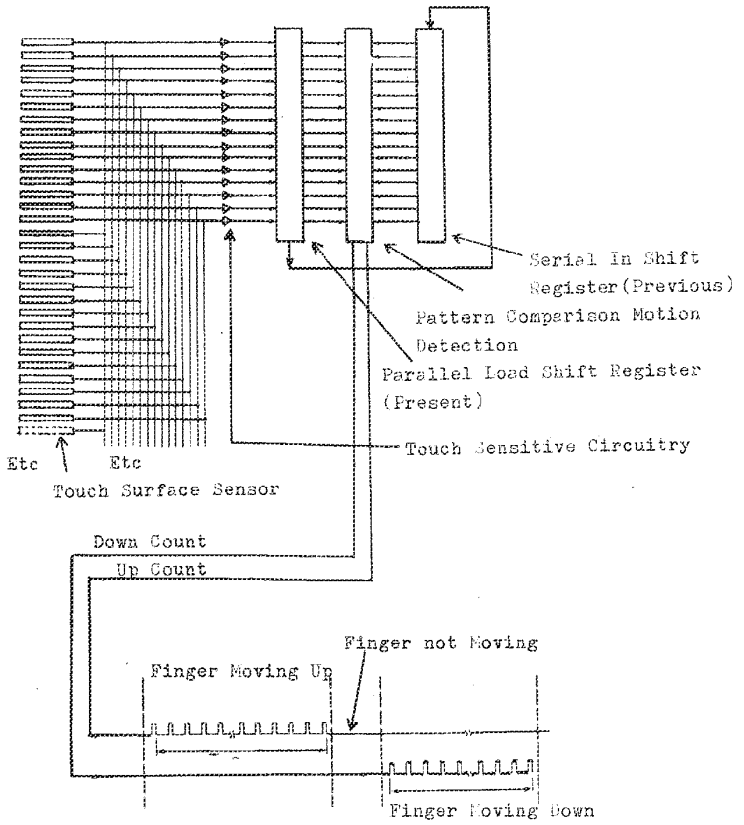


Fig. 2.5 Block and Time Diagram of TASA Touch Tablet

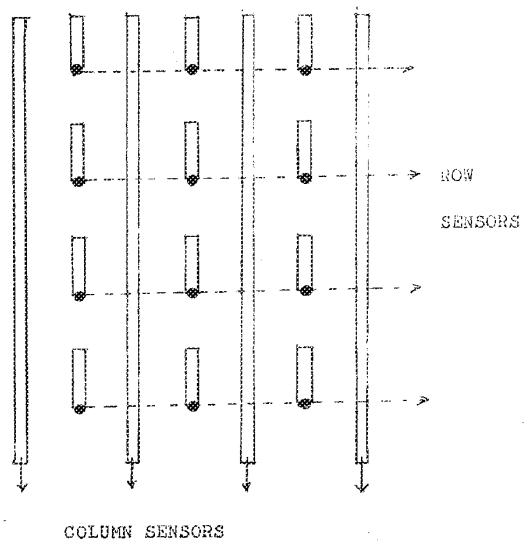


Fig. 2.6 Capacitive Sensor Configuration
of the U of T Touch Tablet

clearly distinguishes at least two points as shown (a) and (b) in Fig 2.7. At least two additional axes are required to distinguish two points without ambiguity if an upper limit to the number of touches is not assumed. Cases in which the number of touches is more than two are shown in (c) to (g) in Fig. 2.7. But as soon as the number of axes is increased, the attraction of any scheme of 2-d projection diminishes because the cost of implementation rises greatly. For example the capacitive tablet may require an unimplementable number of wire layers whereas introduction of sensors and sources for IR and Ultrasound schemes may be extremely difficult. Moreover as the number of the touch points increase, additional axes do not help in resolving a basic shortcoming of the sensor from which only on and off information is available. This is the existence of a region for which identification of points inside is not possible as shown Fig. 2.8.

The video technique seems to solve all the problems embedded in all of the "projective sensor" systems including pressure if "area of touch" by a compressible finger corresponds to the pressure. But a video system is quite bulky due to the optical enclosure and it is slow because it has to access the data stored in memory by the camera processing unit. The maximum speed is limited by the scan rate of the camera(30 per second) even though this maximum can be achieved only by pipelining all the processes required.

Detection of touch by row sensors (a), (b) and column sensors (c), (d) can be interpreted in 7 possible ways as shown below.

Fig. 2.7 Possible sets of points whose existence may be implied by two sensors on both row and column.

Fig. 2.8 Concave touch points that block the points inside. Any point or group of points within the shaded region cannot be identified by any number of projections.

2.4 APPROACHES TO THE PROBLEMS

One concludes from the previous discussion that existing systems and devices do not provide an appropriate means to reach our goal. The major problem of the projective sensor is generally the ambiguity on multiple points. As a solution to this problem, multiple axes are required. This however, increases the cost as well as the number of points to be scanned. Furthermore it cannot eliminate the ambiguity within regions of "concave touch points" using any reasonable number of axis. On this basis, all projective methods must be discarded.

One idea of some significance that can be introduced is to avoid scanning all the pixels in the tablet which contain no information. For example, scanning all 2048 points of a tablet having a resolution 64 by 32 for fewer than 10 points is really quite a ridiculous approach. In fact, if the number of the points to be searched is comparably small, then an improved algorithm, here called binary scanning, can be used. It is described as follows.

Consider a plane of a tablet with resolution 8 by 8 to be searched for a touch point as shown in the Fig. 2.9. First, check the tablet for touch as a whole region as shown by the area ABCD in the figure. If touch is detected, divide the tablet into two equal regions shown by the line EF and check each of the two regions ABEF and EFCD for touchedness.

Select the touched region, region EFCD in this case, and divide this into two equal regions as shown by the division line GH, selecting the touched region. Continue this process until no further division is possible, that is, until a unit sensor, designated as the region PKNO in Fig. 2.9, is reached. The figure also shows the sequence of subdivision in the binary scanning operation. More details of this algorithm are given in chapter 4.

Using this algorithm, a search for one point on a tablet having a resolution 64 by 32, requires twenty two scanning times:

$$2*\log_2(64*32)=22$$

If there is no overhead in binary scanning and scanning begins at the "top of the tree" (that is, with a region in which all pixels are grouped together), then using binary scanning, the number of touched points that can be identified in the time that it would take to detect one touch if all pixels are scanned one by one linearly is

$$N=\frac{64*32}{22}=186.$$

This shows immediately that the binary scanning method is much superior to linear scanning if the number of points to be scanned is fewer than 186.

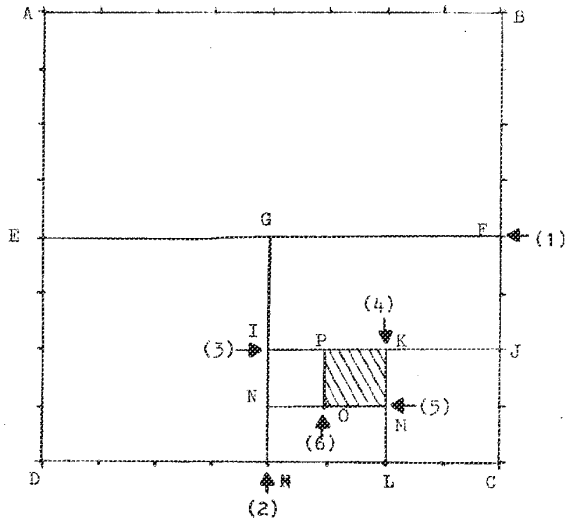


Fig. 2.9 Binary scanning operation.
(n)-Sequence of subdivision in binary operation.

For example the speed gain over linear scanning when the number of points is ten is

$$\frac{64*32}{22*10}=9.3$$

That is, for 10 points binary scanning is about 9.3 times faster than linear scanning.

Now compare such a 64*32 binary scanning tablet with a 2-d projective touch tablet such as used in the HP Personal Computer Input Screen, or with the capacitive single touch tablet with 64 by 32 resolution. In each case the speed gain for detection of a single touch is

$$\frac{64+32}{22}=4.36$$

or about 4.3. That is, the binary scanning tablet is potentially 4.3 times faster than one for which only sequential scan is possible.

Even if the binary scanning algorithm is applied to the projective tablet, the speed for the projective tablet will not exceed the speed of the 2-d image tablet. The scanning time for a projective tablet having the same resolution is found as follows; Since there are 64 + 32 sensors only, and the binary scanning algorithm is applied to the row and the column sensors separately, the scanning time is

$2*\log_2(64)+2*\log_2(32)=22$

Thus the binary scanning algorithm seems to be a very attractive one for application to the FMTSID. However it necessitates two special hardware features:

— 1. Unlike the projective sensor system which allows to address only a group of positions in a column or a row, all individual positions of m by n tablet should be addressable.

— 2. It should be able to group a number of adjacent sensors as one larger sensor region.

The first requirement is to permit sensing of multiple touches, the second allows for the binary scanning algorithm. In addition to these necessary features, a measure of the intensity of touch should be considered as a third requirement to increase the capability of the FMTSID.

The sensors that can possibly accommodate these requirements are many. However the degree of difficulty in implementation varies one to another. In the following, various sensor types are examined in view of the requirements identified above.

A resistive image sensor can be used with a slight modification of the multiplexor as shown in the Fig. 2.1,

however far too much time is required to evaluate the resistance using two port parameter calculations. Besides, the basic approach requires that all the row resistances have to be evaluated once using linear scanning; that is, binary scanning is not applicable to rowwise scanning.

For the video system, the row and the column scanning registers are not accessible and with the video data stored in memory, it is not possible to satisfy the second requirement. However the third requirement may be met by the addition of a little more hardware. In general, therefore, it is not a good choice.

None of the other devices presented in the previous section satisfy the requirements. Thus it is necessary to identify further sensor types.

A resistive polymer (J.S.R) was examined [ref 2.7]. The polymer changes resistivity with applied pressure. One of the applications shown by J.S.R has two closely-placed, but separated, metal coated plates on the PC board on which the rubber sheet lies such that if the rubber is pressed down, the two plates are connected. This unit may be used with a lot of multiplexors to select one of all sensors on the board. However a problem with this sensor is that it takes a very long time for the material to recover electrically to the original state after the finger is released (about 100 msec).

Piezoelectric material was also examined. One possible approach is to apply the same hardware technique used in keyboard applications. However this approach does not eliminate the multiplexing problem. Unfortunately during the time required for multiplexing, the charge developed by the impact of the touch can be lost.

Finally, it seemed that the capacitive sensor offered the greatest potential of all available approaches for the following reasons:

— 1. Capacitive sensors do not need additional equipment, that is, the basic insulating sheet and touch plate are sufficient.

— 2. Multiplexors for individual sensors can be avoided or are degenerately simple using row and column addressing methods described in chapter 3

— 3. Capacitive sensors can accommodate all three features identified above. In particular, a measure of intensity is available since the area of finger contact and corresponding capacitance increases with finger pressure.

— 4. Capacitive sensors are very durable since no additional elements are needed and there is no recovery

time phenomenon involved as exists with resistive rubber.

However there are some drawbacks to capacitive sensors. First they require time to charge and discharge. However these times can be reasonably well controlled. The second drawback is that a capacitive sensor generates radio frequency noise when the tablet is touched. The third drawback is low resolution. However the low resolution can be compensated by software technique as discussed in chapter 4. Noise in the radio frequency spectrum, which is potentially a problem in some environments, does not deteriorate the operation of the tablet. Since level of the noise could not be known at the time of design, it was not considered as a factor in the choice of sensor.

2.5 CONCLUSION

In this chapter, several kinds of touch sensitive input device have been examined as a part of the process of developing a flexible multi-purpose input device. Many devices and sensor types have been analyzed from different points of view in order to achieve the desired goals.

With respect to the initial goals of developing a flexible multi-purpose input device, it seems that the touch tablet as conventionally defined, does not fulfill the

requirements of flexibility and adaptability, but rather that a fast multiple touch sensitive input device (FMTSID) must be pursued to meet the real requirements.

In order to reach this goal, capacitive sensor based hardware with a binary scanning algorithm was chosen for development.

CHAPTER 3

HARDWARE DESIGN AND IMPLEMENTATION

3.1 INTRODUCTION

This chapter describes the details of the hardware implementation of a fast multiple-touch-sensitive input device (FMTSID). The design of the hardware is based on the hardware requirements identified in the previous chapter and on tradeoffs between software and hardware. The hardware basically consists of a sensor matrix board, row and column selection registers, A/D converting circuits and a dedicated CPU.

The design of the sensor matrix is based on the technique of capacitance measurement between a finger tip and a metal plate. Row selection registers select one or more rows by setting the corresponding bits to a high state in order to charge up the sensors while the column selection registers select one or more columns by turning on corresponding analog switches to discharge the sensors through timing resistors. The intersecting region of the selected rows and the selected columns represents the selected sensors as a unit. A/D converting circuits

measure the discharging time interval of the selected sensors. A University of Toronto 6809 board was used as dedicated CPU.

The details of the sensor matrix design are given in section 3.2, with the rest of the hardware described in section 3.3. Section 3.4 describes the scanning sequence used in conjunction with the hardware development process while section 3.5 concludes with a description of the hardware implementation.

3.2 THE SENSOR MATRIX

The design and construction of the sensor matrix board is straightforward. The touch surface of the sensor board consists of number of small metal-coated rectangular-shaped areas serving as sensor plate capacitors. The design of the metal plate area of a unit sensor depends on the measurable capacitance change that results when the area is covered by a finger tip, and on the resolution that can be implemented.

A 12" by 16" sensor matrix area with a resolution of 32 by 64 was chosen, resulting in 7 mm by 4 mm area for each sensor. The estimated capacitance between the sensor plate and a touching finger separated by 3 mil (0.075 mm) Mylar insulating coversheet is

$$Cs = e\frac{A}{d} = 3 * 8.85 * 10^{-12}\,[F/m]\frac{7mm * 4mm}{0.075mm} = 10pF$$

where e is the dielectric constant of the insulating coversheet, A is the area of the unit sensor and d the thickness of the insulating sheet.

The charge associated with the touch capacitance is stored between the sensor plate and the touching finger acting as ground. For this purpose human beings can be considered to be a large charge reservoir. For the static charge case, a suitable model of a human being is an approximate 100 pF capacitor with one plate connected to ground. [ref 3.1] Therefore, it is safe to assume a touch as ground reference for measurement of relatively small capacitances.

The 10 pF of sensor capacitance change is relatively small but measurable. For a timing resistor of 100 k, the time change due to the sensor capacitance change is about 1 micro-second. The tradeoff between the time taken for the measurement of the capacitance and the ease of measurement seems to be obvious. If the capacitance is high, it is "easy" to measure but it takes longer, slowing down the scanning procedure. The clock cycle used to count the discharging time is also limited by noise in the analog circuits as well as by timing limitations of the TTL circuits used. With these limitations in mind, the period of the counter clock was chosen to be 100 nano-seconds.

In order to select a sensor by row and column access, two diodes were used for each sensor. One diode, connected to the row line, is used to charge up the sensors in the row. It is referred to as the Charging Diode (CD) as shown in Fig. 3.2. The CD also serves to block the charge flowing back to the row line when the row line voltage is dropped to zero. The other diode called the Discharging Diode(DD), connected to the column line, enables discharging of the selected row sensors to a virtual ground. Also the DD blocks charge flow from the sensors in the selected row to the sensors in the unselected row during the discharging period. The selection of rows, by the row selection procedure, causes the sensors to be charged. The sensors in the column are then discharged through associated timing resistors connected to the column selection switches.

Fig 3.1 (a) shows the components associated with a selected sensor. There are two related time periods: One for the selection of rows (that is, the charging period) and the other for the selection of columns (that is, the discharging period). The capacitance is measured while the sensor discharges. The signal output during discharging is shown in Fig. 3.1 (b).

Analytic equations can be derived for the model assuming that the reverse diode resistance is much higher than the discharge resistor R and the forward resistance is much

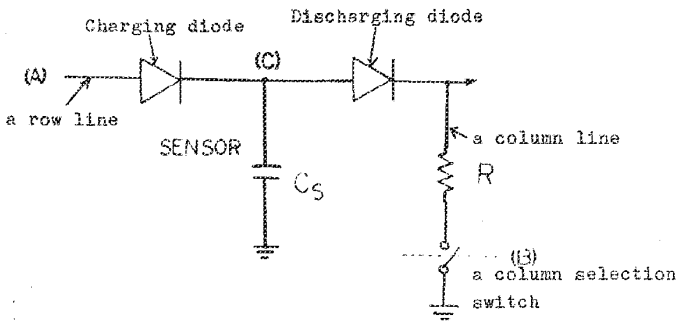


Fig. 3.1 a. A model of a selected sensor in the sensor matrix

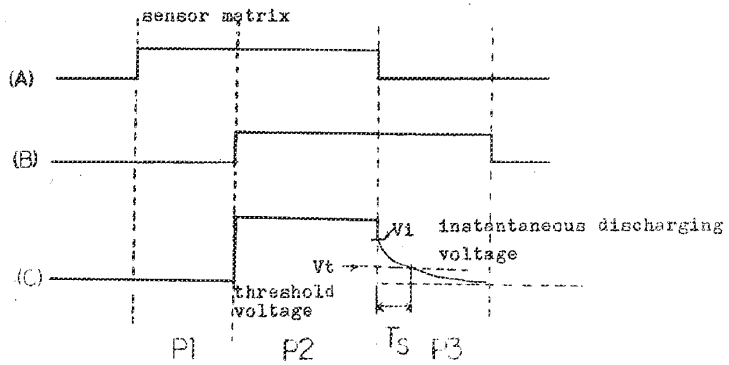


Fig. 3.1 b. The timing diagram for discharging time measurement of a selected sensor as shown above.

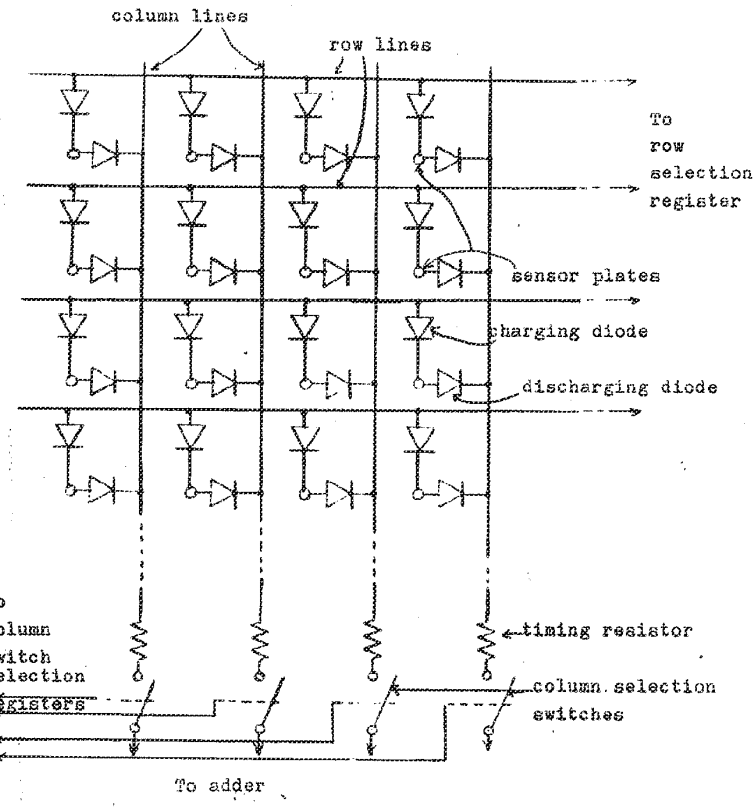


Fig. 3.2 A small section of sensor matrix.

smaller than the discharge resistor R. The derivation is as follows.

The discharging voltage (V) is given by

$$V=Vi*\exp(\frac{-t}{T})$$

where Vi is the instantaneous initial voltage of the discharging period and T is the time constant.

T is given by

$$T=R(Cs+Cr)$$

where Cs is the sensor capacitance and Cr is the reverse bias capacitance of the diode.

The voltage Vi can be found from

$$Vi=(\frac{Cs}{Cs+Cr})(\frac{Qs-Qf}{Qs})(Vcc-Vd)$$

where Qs and Qf are respectively the charges stored in the sensor and the forward biased diode just before the discharging period begins, and Vcc and Vd are respectively the high state voltage for CMOS and the diode voltage drop. In the equation for Vi, the first factor is associated with charging up a reverse biased diode while the second factor results from the charge in the forward biased diode stored

during the charging period.

The discharge time is measured by the comparison with a threshold voltage (Vt) and it is given by

$$Ts = -T\ln\left(\frac{Vt}{Vi}\right)$$

From these analytic equations, the following conclusions relating to the selection of appropriate voltage levels and the diode type can be made.

— 1. The higher the Vcc, the less sensitive is the measurement to the threshold voltage at V = Vt where the Vt = aVcc and the sensitivity,that is the derivative of V with respect to t, is -aVcc/T.

— 2. The smaller the reverse bias diode capacitance, the higher the resulting Vi and the less time it takes to measure the same sensor capacitance.

— 3. The smaller Qf can be made, the higher Vi will be.

The first implication is the choice of CMOS logic to interface the sensor matrix using high logic voltages with Vcc equal to 15 volts. To use CMOS logic directly connected to the sensor board which is prone to high static voltage from touch, the circuitry must be protected. High negative voltage bypass diodes are connected to each row

line(charging source) for this reason.

The second implication is the use of diodes with small reverse capacitance. The diode 1N 4148 was chosen for low capacitance, low cost, and availability. The reverse bias capacitance of this diode is specified as 4 pF.

The third implication relates directly to the forward current of the diode since Qf = Cf * Vd = k * Id * Vd, where Cf is the forward capacitance, Vd is the diode voltage drop, Id is the diode forward current and k is some constant. Furthermore, since Id = (Vcc - 2*Vd)/R, the Id selected is then interrelated with the timing resistor chosen.

Further analysis of the sensor matrix is of interest: there are 2048 such unit sensors implemented on the P.C board. Accordingly the analysis is rather complicated because the rows and columns are electrically not completely separated. A small section of the sensor matrix configuration is shown in Fig. 3.2 for illustration. The reverse bias capacitance couples the sensors in a column. Moreover there exist capacitances between columns due to the physical configuration of the sensor plates and wires and due to the parasitic capacitances in the circuit, and these couple the sensors in a row. The first of these coupling is seen to be unavoidable while effort was taken to reduce the second.

A simple column sensor array has been analyzed as

shown in appendix A. Only the results will be discussed here. The analysis is based on an effort to obtain Vi, the instant initial voltage in the discharge period, r, the ratio of the sensor capacitance over the surrounding capacitance, and m, the separation parameter in the rows.

The instantaneous discharging voltage Vi for a case when a selected sensor is touched, is given by

$$Vi=\frac{st*a+0.5*(n-1)-f*b}{st+0.5*(n+3)}(Vcc-2*Vd)$$

where a = (Vcc-Vd)/(Vcc-2*Vd), b=Vd/(Vcc-2*Vd), f=Cf/Cr, and st=Cs||Ct/Cr.

The ratio of the sensor capacitance to the intrinsic capacitance of the surroundings, r, is

$$r=\frac{st}{\frac{n+1}{2}+st}$$

The separation parameter m is the number of non-selected sensors which must be touched, to cause a non-touched, but selected, sensor to report as "touched". Appendix A shows equations for derivation of this parameter and its evaluation by computer iteration in terms of variables such as the ratio of Cs/Cr and Cf/Cr. When Cs/Cr = 2.5 and Cf/Cr = 12.5, the separation parameter m is about 8, and when Cs/Cr

= 5.0 and Cf/Cr = 12.5 the m is about 25. The first result implies that if more than 8 sensors are touched in a column, then the result will be a wrong report that all the sensors in the column have been touched. The second result implies that if more than 25 sensors are touched in a column, the result will be a report that all the sensors in the column have been touched. Nevertheless, all of the analysis for the single selected sensor model is still valid with some complication.

The analysis becomes more complicated when the parameters for rows and column are included in the equations. Since the operational amplifier adds currents from selected columns and consequently the discharging time increases by 1+ln(2)/a, where a = ln(Vi/Vt), when the number of columns increases by factor of two, the reference values are expected to be increased by the factor of 1.57 for a = ln(10/3) = 1.2 correspondingly. However, an increase in the number of rows in a group is not expected to increase the reference values because the charge stored in the non-selected reverse bias diodes becomes smaller whereas the charge stored in the selected forward bias diodes remain constant during charging period and consequently Vi becomes smaller.

Efforts have been made to separate the columns as much as possible. However there still will be some coupling capa-

citances between columns due to physical adjacency; but their effects are considered to be minimal.

3.3 INTERFACE CIRCUIT DESIGN AND IMPLEMENTATION

Interfacing between the sensor matrix and the dedicated CPU requires three main circuit blocks: row selection registers, column switch selection registers and A/D converting circuits. The CPU selects the row or rows of a sensor group, initiating charging of all the associated sensors. After a charging interval, the CPU discharges the selected column or columns corresponding to a sensor group by connecting a group of discharge resistors whose current is summed via a high slew rate operational amplifier.

There is tradeoff between the scheme using a single bit for each row and one using decoding circuits to implement binary addressing suited to the binary scanning algorithm. The bit per row scheme requires 32 bits of register while binary coding needs 6 bits of register since only 32 * 2 -1 patterns of sensor grouping exit according to the binary scanning algorithm. However, the first scheme was chosen because it is ultimately flexible, that is, it allows one to implement "all" scanning modes by means of software. Thus changes can be easily made in the case of difficulty with implementation of a particular software algorithm. The

implemented prototype uses four 8 bit registers with a common reset signal for row selection.

For the same reasons sufficient column switch selection registers are provided so that one can generate any group pattern through software. However, to reduce the number of chips on the board, four switches are controlled by each bit resulting in four simultaneous analog signals and necessitating four counters. However as a result only 2 8-bit latches are needed to control 64 switches. The four sets of data are provided to the CPU after each scan. These data are manipulated by software in correspondence with the selected resolution mode. This implies the hardware itself acts as an array of 32 by 16 bits while the software emulates 32 by 64 bits of scanning resolution. The scanning algorithm is explained in chapter 4.

The charges stored in the selected row(s) flow down through the selected switches to the virtual ground of the fast operational amplifier. All the discharging currents are correspondingly added to produce a signal from which the discharging time of all the selected sensors can be found by comparison with a threshold voltage. The output of the negative voltage comparator is fed to the enable signal of the counter and to the data bus as a status bit for the counter readiness. All the counters are reset when the row selection registers are deselected in order to initiate the counting

greater number of bits is found to be necessary. The 8 bit counter enables the measurement of a 7 bit capacitance change regardless of the degree of overflow in the counter. This means that if the difference, touched to non-touched, does not exceed 127, the difference can be obtained without ambiguity. For example, a counter value 4 is larger than f8 if the difference is less than 7f in hexadecimal; in other words, an unsigned 8 bit comparison would not be appropriate. Therefore the counter does not need to accommodate the entire discharging time including the time due to the surrounding capacitance, which may require more than 8 bits. Some manipulation is done in software to utilize the facts above. A complete analysis is given in appendix B.

The remaining circuits are for shifting CMOS levels to TTL level and for decoding addresses. Eight addresses are decoded with a R/W signal. One write address is used for a single register whose output controls the discharge of all the sensor matrix at once, the grounding of the ground strip on the board for template recognition applications as described below, and as well as controls three LED outputs.

A metal strip is located on one side of the touch surface of the board. This strip is a programmable ground strip which can be grounded to allow a metal-coated pattern on the back side of the template shown in Fig.3.3 to be scanned. It can be ungrounded for normal binary scanning so

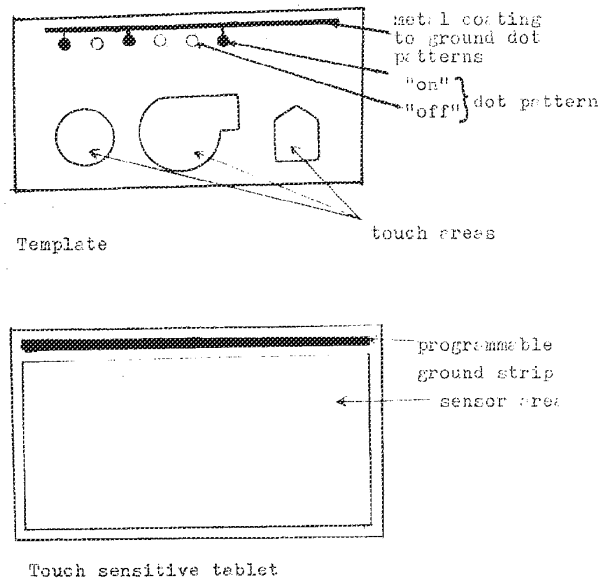


Fig. 3.3 Template and touch sensitive tablet.
The template pattern lies on the sensor area.
Its metal coating touches the ground strip
which is grounded to initiate recognition of
the template.

that the "touch" by the pattern can be ignored. Otherwise it increases the time to scan all the points on the touch surface including the unneeded points from the template because the binary scanning time is directly proportional to the number of points on the tablet.

3.4 THE HARDWARE SCANNING SEQUENCE

This section describes the reading sequence for data from the sensor matrix board. A block diagram of the hardware circuit is shown in Fig. 3.4 to assist the reader. The sequence is as follows.

— 1. Reset the row selection registers. That is, ground the inputs to the sensor matrix.

— 2. Discharge all the sensors directly to ground (but not through R) for about 2 microseconds.

— 3. Select the row selection registers with an appropriate bit pattern.

— 4. Select column switches by turning on the appropriate bit pattern in the column switch selection registers about 4 micro-seconds to stabilize the sensor charge.

— 5. Reset the row selection registers and the the output counters together to initiate counting of the discharg-

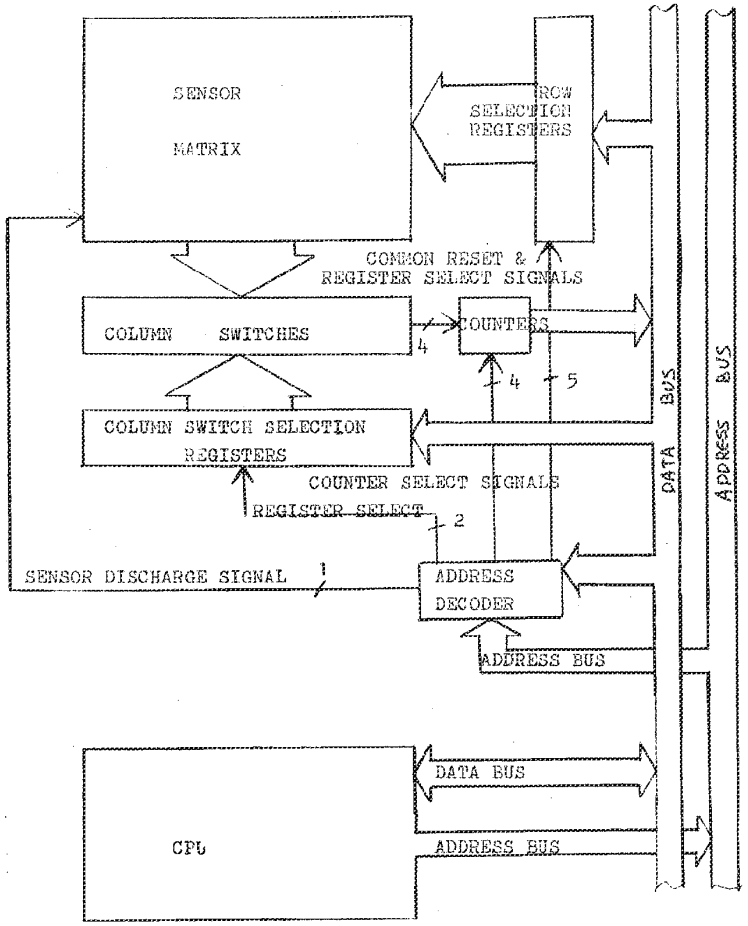


Fig. 3.4 Block diagram of the hardware.

ing time.

— 6. Wait until the counters are ready by reading the status bits.

— 7. Read the counters.

The bit patterns in the row selection register and in the column switch selection register represent the row and column addresses respectively. These, however, have to be converted to a convenient form corresponding to physical placement which is understandable to the user or to the interfacing routine. This process is described in the next chapter.

3.5 CONCLUSIONS

In this chapter, tradeoffs between software and hardware have been discussed while introducing the design of the sensor board and its interfacing circuits. A prototype tablet has been implemented and tested. There were difficulties but all have been surmounted. The detailed results of performance testing are described in the chapter 5.

CHAPTER 4

ALGORITHMS AND SOFTWARE STRATEGIES

4.1 INTRODUCTION

In this chapter, the software tasks to be performed and details of the algorithms developed are described.

The first task is that of grouping sensors such that a variety of scanning algorithms are allowed so that the flexibility of the hardware is not lost. Definitions of various resolution modes and sensor grouping structures are described in section 4.2. In section 4.3 the address mapping between the physical location and its reference memory address, and bit patterns of the row and column selection registers for all possible groupings of sensors, are defined.

The second task is to reduce the digitization noise of the discharging time, and to identify a touch on the sensor matrix using the threshold method. These processes are described in section 4.4. Section 4.5 details the scanning algorithms. In section 4.6, a few interpolation schemes are described whose purpose is to compensate for the low resolu-

tion of the hardware.

Employing these algorithms and tasks, section 4.7 indicates the programs in the host computer and the dedicated CPU are organized. Finally this chapter is summarized in section 4.8.

4.2 SENSOR GROUPING STRUCTURE

In this section, various groupings of sensors in both column and row directions are defined in such a way that the implementation of most algorithms and software strategies is permitted. Corresponding bit patterns in the column and row selection registers are established for a number of groupings of sensors and their physical locations. From the hardware standpoint, there are many possible groupings of sensors that are appropriate. However, there is limitation to implementing all possibilities because of the corresponding reference value storage requirement.

Amongst many possible ways to group the sensors, one that permits easy implementation of all the algorithms in the dedicated CPU, is a rowwise or columnwise binary tree structured grouping. This grouping allows groups of 1, 2, 4, 8, etc in row or in column. Fig. 4.1 shows the grouping scheme in a binary tree structured way for sensors in an 8 by 8 tablet. In order to describe the bit patterns and

reference value data structures in a later section, a notation that represents a unique group of sensors on a tablet is necessary. In support of this notation the coordinates of column level and address, row level and address are introduced as shown in Fig. 4.1.

Level 0 implies that all sensors are grouped into one row or column, Level 1 represents grouping all sensors in a row or column into two equal parts. Level 2 implies grouping all sensors in a row or column into four equal parts. The position of a group of the sensors in the same level, is described by a corresponding row or column address. The relationship between row/column address and level is shown in Fig. 4.1.

For example, row level = 2, row address = 1, column level = 3, and column address = 3 describes the group or sensors in the shaded region (A) in Fig. 4.1. The group of sensors in the region (B) is given by the coordinates (column level, column address, row level, row address) = (1,1,2,3).

This grouping structure, however, does not allow arbitrary location of a group of sensors on the tablet; even though the number of sensors can be grouped by 1, 2, 4, etc, groups cannot overlap each other in a row level and column level other than their own. Otherwise the grouping structure provides many useful features. For example the

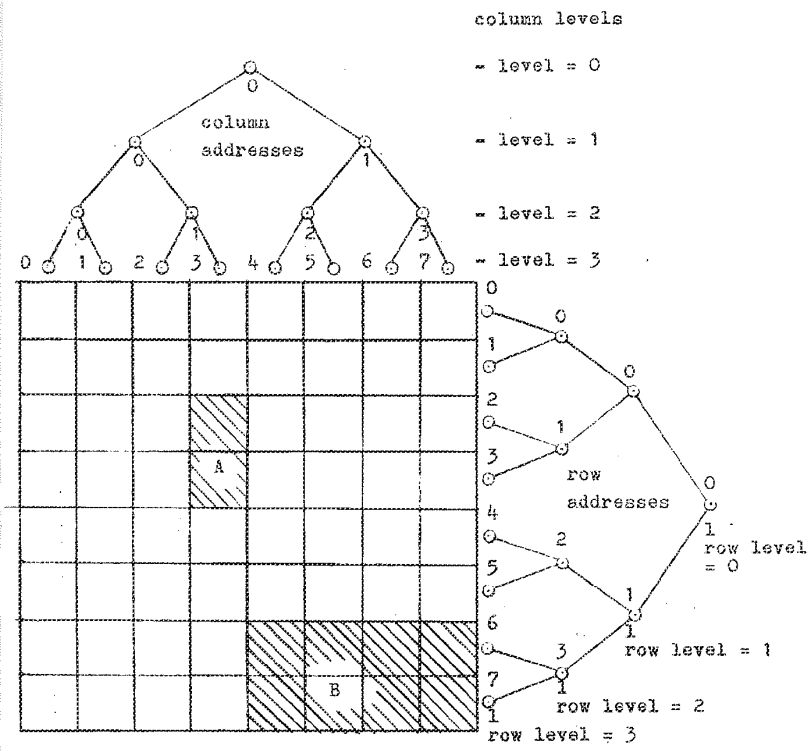


Fig. 4.1 Grouping the sensors in two binary-structured ways for an 8 by 8 sensor tablet.

A group of sensors is described by 4 coordinates (column level, row level, column address, row address). For example, the sensors in the shaded regions A and B are designated by (3,2,3,1) and (1,2,1,3) respectively.

groupings needed for the binary scanning algorithm are shown in Fig 4.2 in order to display the relationships between the nodes in the "tree" and the corresponding physical location on the tablet. This arrangement utilizes a part of the grouping structure implemented.

4.3 BIT PATTERNS IN THE SELECTION REGISTERS AND ADDRESS MAPPING

The bit patterns in the row and column selection registers are directly related to the groupings of the sensors. A group of sensors corresponding to (column level(cl), column address(ca), row level(rl), row address(ra)) have a unique bit pattern in the row and column selection registers. It is given by

bp = a(0) a(1) a(2) a(3) a(4) . . . a(i) . . a(n-1)

where a(i)=1

for i={ i | num_bit * ra(or ca)
<= i < num_bit * (ra(or ca) + 1) }

otherwise a(i)=0.

And where the num_bit is defined to be

$2^{(maximum\_number\_of\_levels - rl(or\ cl))}$

Fig. 4.2 Grouping of an 8 by 8 sensor array for the binary scanning algorithm. Each node in the "2-d tree" represents a grouping of all the sensors or nodes below.

and n is the number of sensors in a row or column. The total number of grouping levels of a tablet with resolution (r by c) is given by

$$\lceil 1+\log_2(r) \rceil * \lceil 1+\log_2(c) \rceil$$

In particular, for the prototype, the number of grouping levels is 42, that is

$$\lceil 1+\log_2(64) \rceil * \lceil 1+\log_2(32) \rceil = 42$$

In other words, there are 42 different sensor groupings

Since the total number of nodes in a "perfectly balanced" binary tree with bottom level number m, is given by

$$(2*2^m-1),$$

the total number of combinations of nodes in the row and column, having bottom levels m and n, is given by

$$(2*2^m-1)*(2*2^n-1)=4*2^{(n+m)}-(2^m+2^n)+1$$

Thus since each combination of row and column nodes requires a reference value, the storage required for the total grouping structure is almost four times as great as the number of sensors on the tablet. This results in a storage for reference values in the prototype of about 8k bytes.

Reference memory address mapping is shown in the Fig 4.2. The reference address is obtained from the row and

column addresses and the row level and column level values. The conversion from one to the other is very simple in assembler and access is more rapid than if pointers were used.

The masks for all grouping levels in both row and column addressing, as shown in Fig. 4.3, use bit stuffing in front of the row and column addresses to identify a group of sensors uniquely at all levels. Since only one more bit is needed to represent the addresses of any node as well as any leaf in a tree, the number of address bits required by the reference address generation scheme shown in Fig. 4.3 for a tablet of resolution m by n, can be seen to be

$$\log_2(n)+\log_2(m)+2$$

For the prototype whose resolution is 64 by 32, the requirement is 13. Therefore, the address mapping scheme uses a full 8k spaces whereas the requirement as provided above is for a little less than 8k bytes. Nevertheless, it is very close to the maximum usage and the scheme is very efficient.

4.4 NOISE REDUCTION

The capacitance values vary with spatial arrangement as well as with grouping level. In order to remove the variation on capacitance discrepancy over the spatial arrange-

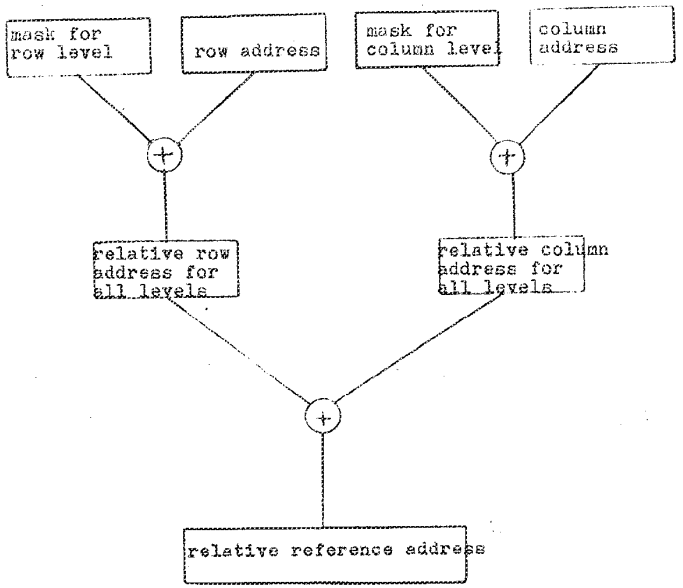


Fig. 4.3 Reference address mapping from a physical address consisting of row and column addresses.

ment, reference values are obtained for the untouched tablet and stored during the initialization period. After initialization the data are then obtained by subtraction of the reference value from the current value. This process eliminates the spatial noise embedded in the sensor circuits.

The value obtained from a group of sensors still contains various time dependent noise signals originated in the analog circuits as well as the noise corresponding to variations in touch. Therefore it cannot be used directly. To reduce the effects of time varying noise, the CPU reads the same group of sensors many times and a average is used. The process of averaging involves the overflow of an 8-bit register as described in appendix B. The averaged datum is then compared with a threshold for its row and column levels to detect touchedness. The expected level of noise varies with the grouping level. Therefore each combination of row and column levels has a threshold. The thresholds are obtained either through direct input from the user or by a process of automatic threshold setting which is performed during the initialization period.

Another approach to reducing the noise is a scheme relying on spatial arrangement. This method is applied only for the interpolation of a number of touched sensor values. The basic idea is that pixels remote from the touched sensor can be weighted to reduce the noise to signal ratio.

4.5 SCANNING ALGORITHMS

Three scanning algorithms were considered and implemented. They are linear scanning, modified linear scanning and binary scanning. With the linear scanning algorithm, the CPU scans the tablet one physical sensor at a time. The modified linear scanning algorithm is based on the fact that the tablet can be configured as to emulate a projective sensor. First the CPU searches for a touched row and, then only the touched row is divided into many columns for detecting individual sensor locations. The binary scanning algorithm has been introduced in the chapter 2. The details of the scanning processes in C language description follow.

```
touched(difference)
int *difference; /* return value of the scanning a pixel */
{
current = readdata();
*difference = current - reference;
if (*difference > threshold)
   return ( true );
else
   return ( false );
}
```

linear scanning

```
ls(levelr,levelc)
int levelr, levelc; /* input to define the resolution */
{                   /* levelr/c = row/column level    */
int limitcoladr, limitrowadr, coladr, rowadr;

limitcoladr = 2 << levelc;
limitrowadr = 2 << levelr;
for ( coladr = 0; coladr < limitcoladr; coladr++)
for ( rowadr = 0; rowadr < limitrowadr; rowadr++)
  {
  rowbitpattern = getbitpattern(levelr,rowadr);
  colbitpattern = getpattern(levelc,coladr);
  reference = getreference(levelr,levelc,rowadr,coladr);
```

```
    if ( touched(&z))
       report(rowadr,coladr,z);
    }
  }
```

modified linear scanning

```
mls(levelr, levelc)
int levelr,levelc;
{
limitcoladr = 2 << levelc
limitrowadr = 2 << levelr

for(rowadr=0; rowadr < limitrowadr; rowadr++)
{
 threshold = getthreshold(levelr,0);
 colbitpattern = getpattern(0,0);
 rowbitpattern = getpattern(levelr,rowadr);
 reference=getreference(levelr,0,rowadr,0);
 if (touched(&z))
 {
   for(coladr=0; coladr<limitcoladr; coladr++)
   {
     threshold = getthreshold(levelr,levelc);
     rowbitpattern = getpattern(levelc,coladr);
     reference        =         getreference(levelr,levelc,
rowadr,coladr);
     if (touched(&z))
       report(rowadr,coladr,z);
    }
  }
 }
}
```

binary scanning

```
bs(coladr, rowadr, levelr, levelc)
int rowadr, coladr, levelr, levelc;
{
rowbitpattern = getbitpattern(levelr,rowadr);
colbitpattern = getbitpattern(levelc,coladr);
reference=getreference(levelr,levelc,rowadr,coladr);
if ( touched(&z))
 {
 if((levelr==limitlevelr)&&(levelc==limitlevelc))
   report(rowadr, coladr, z)
 else
   bs(coladr<<1, rowadr<<1, levelc+1, levelr+1);
 }
else
 {
 if((levelr!=limitlevelr)||(levelc!=limitlevlc))
```

```
    for(i=0; i<3; i++)
    switch(i) {
     case 0: bs(coladr+1,rowadr,levelr,levelc);
             break;
     case 1: bs(coladr,rowadr+1,levelr,levelc);
             break;
     case 2: bs(coladr+1,rowadr+1,levelr,levelc);
             break;
     default: break;
     }
   }
}
```

The binary scanning algorithm is actually implemented in assembler in a non-recursive way; however the structure and the algorithm have not been changed from the representation above. This algorithm and the modified linear scanning algorithm are very sensitive to the values of the thresholds for each level. Thus detecting very small contact areas over a large sensor group area is rather difficult. Accordingly if the threshold for the top level is high then a small intensity of touch may not be detected; whereas if it is low, then the search may not be successful. Consequently this would slow down the scanning procedure since a large number of unsuccessful trys at the bottom level (caused by low threshold in the higher levels) delays the scan of the actual touch points.

4.6 COMPENSATION FOR LOW RESOLUTION HARDWARE

It may seem that the resolution of the hardware is too low for use in graphics applications. However touch

intensity and multi-touch sensitivity can be used to enhance resolution. This is possible because the center of a touch can be most accurately estimated by an interpolation utilizing the values of the adjacent sensor intensities. For a simple example, consider the estimation of the center of the touch by an interpolation method as follows:

Suppose the touched point is (i,j) and its intensity value is z(i,j). Let the dx and dy be the relative position of the interpolation to the integer position (i,j). Then the dx, and dy can be obtained from the values of the adjacent intensities as follows.

$$dx=\frac{\sum_{k}(-1,+1)(z(i+k,j+1)-z(i+k,j-1)}{\sum_{k}(-1,+1)\sum_{\ell}(-1,+1)(z(i+k,j+\ell))}$$

$$dy=\frac{\sum_{k}(-1,+1)(z(i+1,j+k)-z(i-1,j+k))}{\sum_{k}(-1,+1)\sum_{\ell}(-1,+1)(z(i+k,j+\ell))}$$

Thus the estimated center of touch is (i+dx, j+dy).

A simple case is shown in Fig 4.5 where possible interpolation points are shown when 1, 2, 3, 4, or 5 bits of intensity are provided from two adjacent sensors. The picture shows that interpolation points are not evenly populated and that the scheme does not give good resolution even if a fairly high number of intensity bits are provided. Therefore it may be good idea to map each interpolation point to another domain which gives evenly populated points

Fig. 4.5 Centers of pressure interpolated from two sensors with the number of available bits for pressure being 1, 2, 3, 4, and 5 from left to right.

Each line represents a possible interpolation point.

for possible combinations of the intensities from two adjacent sensors. However such an interpolation mapping scheme has not been implemented due to complications involved for more than two sensors.

However the interpolation should be performed in terms of the physical center of the touch shape. Since the intensity given by the capacitance measurement is dependent of the area of touch on the cover of sensor plate, the accurate center of a touch is more dependent of the size of the touch relative to the size of the sensor area as well as the shape of sensor plate.

For the implementation, however, it uses direct interpolation scheme for a few cases. One of interest is interpolating 3 by 3 sensors with a touched point in the center and the other is interpolation of all points on the tablet. The later one obviously gives highest resolution but it simply emulates a single touch tablet with a high resolution.

The software in the dedicated CPU utilizes the communication with the host computer to accommodate the interpolation scheme in the host computer.

4.7 SEQUENCE OF OPERATIONS

The programs in the dedicated processor are sequenced

as shown below. First, the tablet is initialized and a red LED on the tablet is turned on to indicate "DO NOT TOUCH". During this period, the reference values for all groups of sensors and the thresholds for each level are found. Automatic threshold settings, however, do not satisfy the user when the user want to have less sensitivity on the level of touch. Therefore the threshold may be changed by the command from either the host computer or the terminal input. After the initialization, the CPU waits for further instructions from the input device. Whenever it is ready, the input device sets the operation modes through a sequence of instruction(s). The operations of each modes and commands are listed in appendix C.

— Initialize

  - read reference values
  - obtain thresholds

— Send ready signal

  - wait for further instructions

— Set operation modes from input commands

  - set scan modes
  - set resolution (for only linear scanning mode)
  - set communication mode

— Scan and Report

In general, the program in the host computer would be application software that utilizes the FHTSID. The sequence presented below would greatly vary with each application. However for each setting up the tablet for best performance according to the application is essential. The configuration of the system in the software is shown in Fig. 4.4.

In appendix C, the names of the routines and modes of operations in the dedicated CPU, as well as the communication protocols with the host computer, are detailed.

In order to run a program in the host computer, the local terminal inputs are directly transferred to the host computer which makes it possible to communicate between the user and the host computer.

— Signal to the local terminal that it is ready to receive data from the dedicated CPU.

— Wait for an acknowledgement.

— Setup output device.

— Setup the tablet according to the needs

— Read inputs from the dedicated CPU on the tablet.

— Process the data from the tablet.

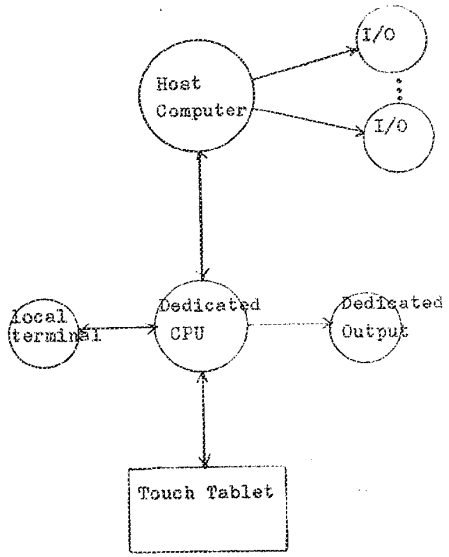


Fig. 4.4 System configuration.

--- Output to the designated device

4.8 CONCLUSION

In this chapter, software strategies and algorithms have been described. All scanning algorithms and commands have been implemented and tested. All software is fairly well documented for the user who wants to implement more features for his own application purposes. Testing results and some demonstration programs for a particular application are provided in the next chapter.

CHAPTER 5

PERFORMANCE TESTING

5.1 INTRODUCTION

This chapter presents the use of the PHTSID prototype as an integrated system, the characterization of the sensor matrix and various aspects of performance testing.

In section 5.2, the sensor matrix is evaluated in terms of its threshold values which are directly related to noise levels present. The times required to measure specific groups of sensors are identified by the reference values for each group. In section 5.3, the use of the prototype having an I/O terminal and a Unix port connected to the dedicated CPU is presented. Section 5.4 focuses on the spatial resolution and locatability of a fine resolution point obtained from interpolation of multi-finger touches. Section 5.5 deals with the test of response time delay under various conditions. In section 5.6, typical applications of the prototype are discussed and finally, section 5.7 summarizes various performance tests.

5.2 SENSOR MATRIX EVALUATION

An ideal sensor matrix for FHTSID would be one that has uniform and small reference values over a grouping level, a large variation of intensity due to a touch and fast measurement time. The sensor matrix of the prototype, however, has a relatively wide range of reference values as shown in table 5.1. However these values, obtained in the normal laboratory environment, do not change very much over extended periods of time. The results show that a decrease of one column-grouping level increases the reference value by a factor of about 1.4. This corresponds well to the estimates made previously in chapter 3. As well the results show that a decrease of one row-grouping level, in contrast, does not increase the reference value in general, even if the number of the sensors is doubled in a group. These phenomena have been discussed in the previous chapter.

From the table, for the tablet left untouched, the capacitance discharge time can be derived directly: 1 count represents a 100 nano second interval.

The thresholds given in table 5.2 represent the stability, over time, for a specific grouping level. Using these threshold values, the CPU does not report untouched points wrongly over intervals of at least 3 hours in both linear and binary scanning modes. The binary scanning mode uses 6 different thresholds, consequently it is very unlikely to

report a wrong point whereas the linear scanning mode uses only a single threshold. Regardless of this, however, the tablet in linear scanning mode has remained for at least 3 hours before reporting a touch when in fact it has been left alone.

TABLE 5.2 TYPICAL THRESHOLDS FOR ALL GROUPING LEVELS

col level		0	1	2	3	4	5	6
row level	0	2	4	3	4	4	4	3
	1	2	3	4	5	6	4	3
	2	10	4	4	4	6	4	3
	3	7	4	5	4	6	4	3
	4	8	5	5	5	5	4	3
	5	6	5	5	6	6	4	4

The intensity of a single touch for a grouping level (5,6) varies over the tablet but usually ranges from the threshold value to 15. For the grouping level (0,0), it varies from person to person but it ranges from the threshold to 124. This maximum is obtained when a palm rather than a finger touches the tablet. Another interesting aspect of the grouping level (0,0) is that its scanning is very fast and as well allows detection of hands merely in the vicinity of the tablet.

In order to test the separation parameter as discussed

TABLE 5.1 REFERENCE VALUES FOR ALL GROUPING LEVELS

column level		**0	1	*2	3	4	5	6
row level 0	(a)	77	206	31	24	20	9	4
	(b)	77.0	209.5	34.0	24.5	21.0	10.5	5.3
	(c)	77	213	36	25	23	12	6
1	(a)	16	120	5	133	22	10	4
	(b)	16.0	130.8	11.8	142.8	23.8	11.9	6.0
	(c)	16	144	29	163	26	14	7
2	(a)	69	171	53	205	49	22	6
	(b)	75.5	184.1	60.6	208.0	54.3	27.1	13.6
	(c)	81	199	75	214	69	42	25
3	(a)	108	196	75	228	123	60	29
	(b)	114.9	214.4	87.4	236.6	128.7	64.3	32.2
	(c)	124	230	104	247	137	70	36
4	(a)	113	208	88	242	142	71	34
	(b)	123.8	224.8	97.5	247.0	147.3	73.7	36.8
	(c)	132	241	113	254	154	77	40
5	(a)	116	209	91	245	149	74	36
	(b)	126.5	226.2	100.4	250.5	152.8	76.4	38.2
	(c)	137	241	118	260	159	80	41

(a) minimum value of the reference values in the grouping level
(b) average value of the reference values in the grouping level
(c) maximum value of the reference values in the grouping level

* single overflow; for actual count, add 256
** double overflow; for acutal count, add 512

in chapter 3, a long and narrow metal plate was used to cover a number of sensors in a column. Using this test, the number of sensors covered by the metal plate that report touched sensors inaccurately (the separation parameter) varies from column to column, ranging from 11 to 28. The range of estimated separation parameter with various values of parameters was 6 to 25, discussed in chapter 3; a value reasonably within the actual measured range of variation.

The final test performed in characterizing the sensor was radio frequency interference sensitivity. Due to a lack of appropriate test equipment, a small transistor AM/FM radio was used. The radio had 9 volt battery and was set to medium volume with the antenna extended fully (about 18"). Noise from the radio resulted for almost all frequencies in the range 540 to 1600 Khz. It was observed that the radio makes a low constant noise when the tablet is scanning without touch and that a modulated sound results whenever the tablet is touched. However in comparison to the R.F noise produced when a key is entered on an I/O terminal (Lanparscope XT-100), the noise from the tablet when the tablet is touched, appears to be about 2 times greater. For this comparison, the distance between the device and the radio at which the noise disappears, was used.

5.3 USE OF THE PROTOTYPE

For development and test purposes, the environment in which the prototype operates, is shown in Fig. 4.4. With this configuration, three possible communication states to the 6809 have been considered, as shown in Fig 5.0. State A represents communication between the 6809 and a terminal. In this state most debugging commands are accessible and the CPU scans and reports the touched points to the terminal for all three scanning algorithms: linear scanning, modified linear scanning, and binary scanning. State B involves communication between the 6809 and the host computer. In this state binary scanning (with and without the interpolation mode) is available making it possible for the host computer to interpolate 3 by 3 subarrays around a touched point, as well as to interpolate all points for one complete scanning. Finally state C is simply a "wired" mode which transfers the inputs from the keyboard directly to the host computer so that the programs in the host computer can be executed simultaneously with the scanning program in the 6809 in state B.

Commands for each of the states and the data formats for communicating with the host computer are listed in appendix C.

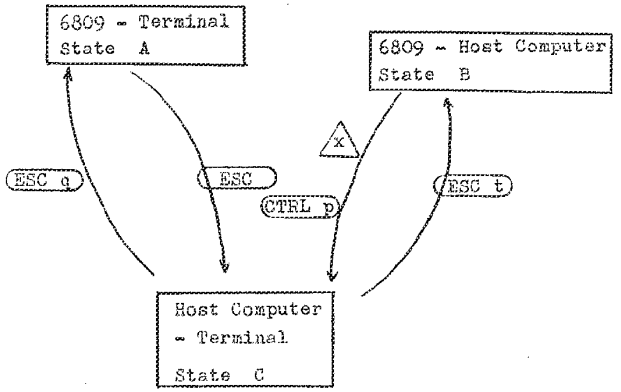


Fig. 5.0 The state diagrm for communication modes

- state change parameter from terminal input

- state change parameter from host computer

5.4 SPATIAL RESOLUTION

One of the benefits that may be obtained from a combination of the multiple sensing and intensity sensing characteristics is interpolation. Many interpolation schemes were considered and the results are discussed in this section.

One possible and immediate interpolation scheme is to interpolate a "touched" point with all adjacent values which may not be large enough to be reported as touched. A local array of 3 by 3 points is used for this interpolation. Some examples drawn on a laser printer using floating point input, are shown in Fig 5.1. These pictures are drawn without feedback, that is, drawn without the operator looking at the output screen. This does not allow the operator to select points where data are sparse in comparison with che intended figure but rather takes direct input from the location of the figure drawn on the input device. The first picture (a) is drawn by moving a finger in a straight line (guided by a ruler) for various angles and the second one (b) is drawn by moving a finger in a line guide by a circle drawn on a template. The third one (c) is picture drawn by putting a whole palm on the tablet. Note that the dots on a line do not have constant intervals and the "empty" location may be reduced by more tries for Fig. 5.1(a). In particular, the dots outside the shape of a palm ,for fig. 5.1(c), are due to the separation parameter. The palm was pushed hard by

the other hand while the CPU scans the tablet. However, they would not be shown for soft touch by a palm.

Since the spatial resolution of the first scheme is limited by the number of bits available from the intensities of an array of 3 by 3 sensors, another scheme was considered. In this scheme, all the points from a complete scan of a tablet are interpolated allowing the potential resolution to be almost infinite. However this process simply emulates a projective device and accordingly reports only single point, which is interpolated from all the points on the tablet. However with this scheme, there could be many ways of pointing to a specific location on a display screen. To demonstrate this, the lines drawn by black dots on Fig 5.2(a) are obtained by moving a finger while a second finger is fixed at various points. The circle Fig 5.2(b) is drawn rather tediously, but with many combinations of finger points. Note, however, that it shows many points outside of the circle line. The pictures drawn by this scheme supposed to be much more accurate to the intended picture on the tablet. But it does not show on the pictures. This is because the operator could not see the output terminal. The locatability of a point on the tablet seen to be good however it should be measured in front of a display terminal for more accurate estimation.

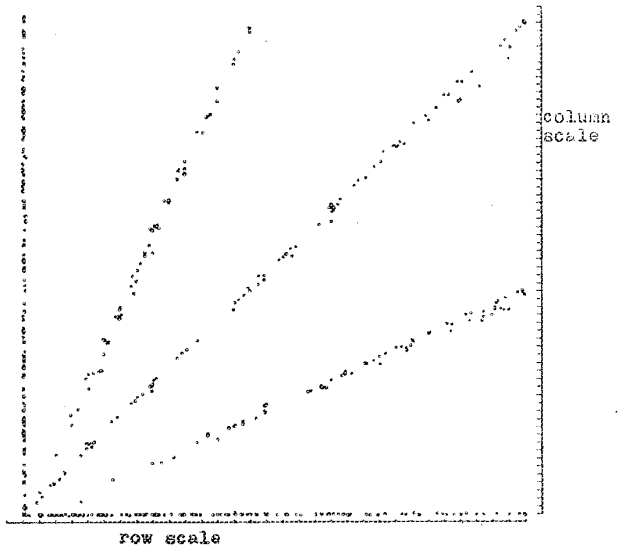


Fig. 5.1(a) Straight lines drawn by the tablet using 3 by 3 sensor array interpolation.

The scales shown represent the boundaries of the actual sensors.

Fig. 5.1(b) A circle drawn by the tablet using 3 by 3 sensor array interpolation.
The scales shown represent the boundaries of the actual sensors.

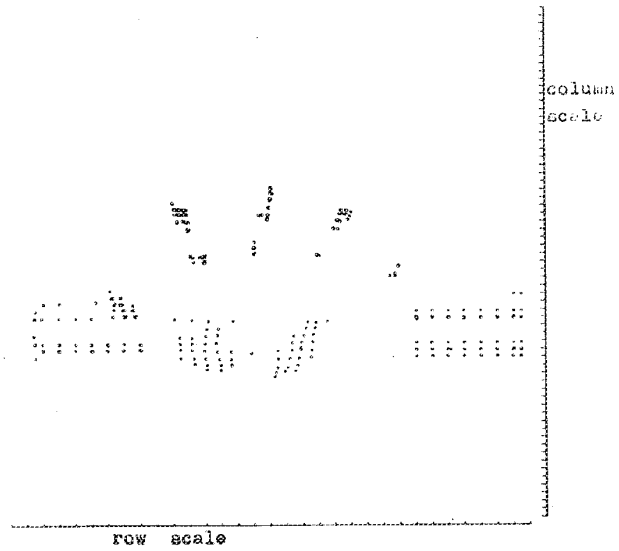


Fig. 5.1(c) A palm drawn by the tablet using 3 by 3 sensor array interpolation.
The scales shown represent the boundaries of the actual sensors.
The dots outside the "palm shape" are due to inability in distinguishing points in a column(separation parameter).

Fig. 5.2(a) Straight lines drawn by the tablet using interpolation of all touched points on the tablet at every scan.
The scales shown represent the boundaries of the actual sensors.

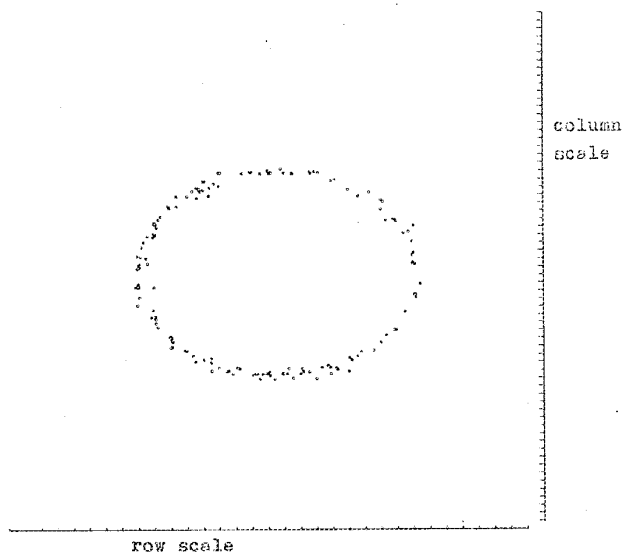


Fig. 5.2(b) A circle drawn by the tablet using interpolation of all touched points on the tablet at every scan. The scales shown represent the boundaries of the actual sensors.

5.5 RESPONSE TIME DELAY

The response time delay is the time delay from the beginning of a touch to an output to either a local terminal or an output device attached to the host computer. For multiple touches, this delay will increase with the number of touches. In general, the response time delay is composed of several different components.

For example, the single touch response time results from the combination of four interrelated processes.

Process I - Generation of bit patterns for row and column registers and reference address calculations from row and column addresses.

Process II - Capacitance measurement from resulting bit patterns.

Process III - Processing for the binary scanning.

Process IV - Processing for communication.

Note that some of these processes are intermixed with others. For example the binary scanning process itself uses capacitance measurement as well as bit pattern generation. However the time implied above for the binary scanning process is just the overhead assuming instant generation of the bit patterns and instant capacitance measurement. In practice each of the processes above can be distinguished

clearly and the response delay time for a single point can be derived from the time required for each.

Various cases are considered and tabulated in table 5.3. Actual response times were measured several times and averaged. These are compared with the results obtained by evaluating execution time of software routines and using the specifications for the communication speed of the terminal. They are tabulated in table 5.4. The results show that the calculations and the actual response time delays lie within a reasonable range, with the overhead due to the command and other uncounted processes ignored in the calculations. Figures are given to show the proportion of the time taken by each processing phase.

5.6 TYPICAL APPLICATIONS

The tablet can be applied to emulate virtually any kind of device that utilizes position input as well as intensity in some form. For example, with appropriate templates, it is possible to emulate a Qwerty keyboard, a Piano keyboard, a percussive device (with low resolution), a cursor positioning device such as a joystick or a mouse, a graphics tablet, or a finger painting input device [ref 5.1]. Above all, it is possible for the FMTSID to form a collection of virtual devices from which input is available

TABLE 5.3 APPROXIMATE TIMES REQUIRED FOR PROCESSING THE SOFTWARE

PROCESS	BEST CASE		WORST CASE	
I	2.3	msec	7.0	msec
II	8.9	msec	28.9	msec
III	1.0	msec	3.4	msec
IV	9.3	msec	10.5	msec
TOTAL	21.4	msec	49.8	msec

I - Time required for bit pattern generation and reference address calculations in searching for a point with binary scanning.
II - Time taken in reading the sensor capacitance in searching for a point with binary scanning
III- Binary scanning overhead; generation of row and column addresses and calling all of above routines.
IV - Communication overhead for a 9600 baud rate assumming 10 bits per transmission for a single character. There are 9 bytes of data for a coordinate.

TABLE 5.4 ACTUAL RESPONSE TIMES

Case	best	typical	worst
(a) pts/sec	17.6	15.2	12.8
msec/pt	56.8	65.8	78.1
(b) pts/sec	19.2	17.2	16.0
msec/pt	52.1	58.1	62.5
(c) pts/sec	24.0	22.0	18.8
msec/pt	41.6	45.5	53.2

(a) measured by one sensor touched continuously
(b) measured by two sensors touched at the same time continuously
(c) measured by four sensors touched at the same time continuosly
Note all values obtained were based on communication between the 6809 and Lanparscope XT-100 terminal using the jump scroll mode.
pts/sec = points per second
msec/pt = miliseconds per point response

simultaneously.

5.7 CONCLUSION

This chapter has included the results of most tests that could be done in a normal laboratory environment. Importantly it has emphasized testing for the "new" sensor with a binary scanning algorithm. It seems, in general, that the new sensor performs better than expected from calculations and simulations. This is because of the unaccounted fact of column capacitance coupling. However some unexpected results were found such as the inability to sense touchedness for three grouping levels (3,0), (4,0) and (4,1). Also noted were strange variations of the reference values in the row grouping levels 0, and 1. These effects likely result from unmatched diodes and other components. More detailed analysis is left for those who are to develop the sensor matrix in VLSI as discussed in the next chapter.

CHAPTER 6

CONCLUSIONS

6.1 INTRODUCTION

A prototype of a fast-scanning multiple-touch-sensitive input tablet having both the adaptability and flexibility for diverse applications has been designed and implemented. Capacitance measurement of individual sensor(s) which can be uniquely addressed using two diodes per sensor, makes it possible to sense both the positions and intensities of one or more simultaneous touches without ambiguity. The sensor matrix is controlled by University of Toronto 6809 board whose serial port is connected to one of the I/O ports of a host computer. Software that utilizes binary scanning for fast scanning an array of 64 by 32 sensors on the tablet, and that communicates with the host computer, has been implemented and tested.

The prototype has, however, still room for improvement in many areas. Section 6.2 focuses on the possibilities for hardware and software enhancement. In section 6.3, the problems of the sensor matrix are discussed and, finally, section 6.4, concludes with directions for future research

on the FMTSID.

6.2 ENHANCEMENT POSSIBILITIES

Even though the prototype has proved that the binary scanning algorithm with suitable hardware solves the problem of speed limitation inherent in measurement of sensor capacitances with linear scanning, some musical input applications, such as to percussive instruments for example, require greater speed with more parameters. At present the tablet operating at the zero grouping level is quite fast, however only a z value is available.

The speed may be improved by both hardware and software enhancements indicated to reduce two major time delays in the response time measurement. These are in the software for the binary scanning process and in communication between the host computer and the device.

A possible enhancement of the software may be derived from the fact that some instruments can be emulated with a particular speed requirement as well as with a particular sensor grouping configuration. In such a case, data structures for reference and bit patterns in the row and column registers could be made specific to the particular application. Thus blocks of sensors can be predefined by software before the application, thereby greatly reducing the

scanning possibilities. Accordingly the author believes that a dynamic data structure with scanning zones on the tablet, which are variable according to the emulation of particular device(s), may increase the response speed considerably.

Hardware enhancements, on the other hand, that increase the speed of the scanning process, should incorporate both parallel communication and possibly a bit slice programmable processor for the scanning process itself. The ultimate response time delay reduction may be obtained not only by reducing the amount of data that communicates between the host and local processors by allowing more processing at the dedicated processor, but also by parallel communication with the host computer. The other possibility is that of putting the capacitance measurement time whose maximum is about 70 microseconds, the bit pattern generation and reference addressing, into a pipeline. This could reduce the time for scanning the tablet completely for the number of finger touches limited to 10, to less than 7.5 milliseconds assuming that longest delay path in the pipeline is the time required to discharge the sensor capacitance. However each of these changes should be considered with a specific application in mind.

6.3 SENSOR MATRIX ENHANCEMENT

There are many problems in the constructed sensor matrix compared to an "ideal sensor". Some are compensated by the software but the others remained unresolved. As discussed in section 2.2 and in section 5.2, the following problems with the sensor matrix remain.

— 1. Each sensor is not perfectly matched due to unmatched diodes, mismatch of circuits attached to the column and row, location differences over the tablet and size variation in the unit sensor plate.

— 2. The threshold for detection of touchedness is dominated by a need for noise elimination rather than "thresholding" a degree of touchedness. This noise originates in the offset voltage of the comparator as well as in instability in discharging currents.

— 3. There are simply too many components to assemble in constructing the sensor matrix;by conventional means, for example there are about 4 thousand diodes to connect(64*32*2).

— 4. The hardware resolution is only 64 by 32 for a 12" by 16" tablet.

Since the diode array on the tablet is uniform, it seems that for large quantity production, a wave soldering technique with automatic insertion equipment[ref 6.1] and small customized diode network chips may be feasible. Some aspects of the first problem and the third problem may be solved by this technique. The second problem may be reduced by a more careful and closer examination of the sensor matrix. The hardware resolution may be increased by a choice of higher dielectric constant separators with smaller sensor area and by increasing the counting clock frequency and using a more stable discharging current.

Current research on flat TV systems in some countries utilizes many techniques for assembly of active components on a large area, as large as a TV screen[ref 6.2]. It shows some interesting results related to the same problem of mounting active components on a large surface area. If the same technology is applied to the assembly of the sensor matrix, it seems possible that the majority of the problems above could be solved. Thus it is likely that the ultimate FMTSID will utilize techniques to be developed by the the large flat screen TV system development in the near future.

6.4 FUTURE RESEARCH

Future research should be directed in accordance with the previous discussion on future enhancements. The author believes that the "final form" of PMTSID should be as a standalone device using a transparent touch plate such as indium tin oxide [ref 6.2]. Such a panel could be put on the top of the display system as is done with current touch screen devices. By this means feedback could be obtained directly under the finger tip. As well of course, tablet templates could be directly and dynamically shown to the user. Such developments, however, must follow more complete research on the use of Thin Film Technology for flat TV screen display systems.

REFERENCES

[ref1.1] "A Touch Sensitive Tablet as Extensible Device" by W.Buxton, G.Fedorkow, L.Sasaki, and K.C.Smith, Computer Systems Research Group, 1983

[ref1.2] "A Touch-Sensitive Input Device"., Proceedings of the Fifth International Conference on Computer Music, North Texas State University, Denton, Texas, November, 1981.

[ref2.0] "Why Touch Sensing" by Louise C.Shaw, Datamation, August 1980, p138-p141

[ref2.1] United State Patent 3,798,370 March 19,1974, Electrographic Sensor for Determining Planar Coordinates and 3,911,215 October, 7, 1975, Discriminating Contact Sensor by George S. Hurst, Elographics, Incorporated,

[ref2.3] "New Touch Screen Computer" by George Mitchell, Computers and Electronics, December,1983, p57 - p66

[ref2.4] Touch Screen Digitizer Data Sheet from TSD Display Product Inc., 1982

[ref2.5] A Flexible Human Machine Interface, M.A.Sc. Thesis, University of Toronto, 1982 by Nimish Metha

[ref2.6] TASA Model: x-y 3600 and x-y controller, Model: FR-105 Data Sheet 1980

[ref2.7] Pressure-Sensitive Conductive Rubber Data Sheet by JSR PCR 1981

[ref2.8] A circuit diagram drawn by Fedorkow in March, 1981, University of Toronto

[ref3.1] "Curing Static Electricity" by Elliott S.Kanter, Radio Electronics, Sept., 1984

[ref5.1] Manipulating Simulated Objects with Real-world Gestures using a Force and Position Sensitive Screen by Margaret R.Minsky, Computer Graphics Volume 18, Number 3 July 1984

[ref6.1] Solders and Soldering by Howard H. Manko, McGraw-Hill, 1979, p252 - p257

[ref6.2] "Liquid Crystals Big, Bright, even Colourful Displays" by Gordon Graff. High Technology may/1984 p55 -p68

APPENDIX A

ANALYSIS ON SENSORS IN ONE COLUMN

In this appendix, sensors in one column are analyzed in terms of the instantaneous voltage during the discharge period, the ratio of sensor capacitance to the surrounding capacitance, and separation parameters. All these parameters are derived from the characteristics of the discharging voltage on the column line.

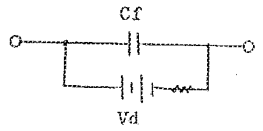
Two assumptions are made in modelling the diodes:

1. The resistance of a forward biased diode is very small compared to the timing resistor R.

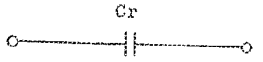
2. The resistance of a reverse biased diode is very large compared to the timing resistor R.

These assumptions simplify the derivation of the equations for all parameters. For these, the diode model is as follows:

forward biased diode



reverse biased diode



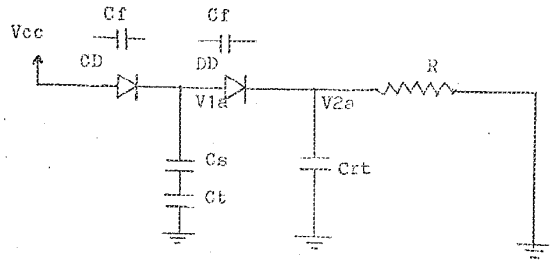
This model is used for all derivations. As well, the following assumptions are made:

1. All diodes are matched.
2. All sensor plate capacitances (Cs) are identical and have a value of zero when the plate is not touched.
3. Each column is electrically separated from all others.

Amongst many possible cases for the analysis of the sensor array, two will be considered here. The instantaneous voltage during the discharge period and the discharge time constant are obtained for both cases. As well, from this analysis separation parameters are obtained for various parameter changes.

Case A: A selected sensor is touched, while the remaining sensors in the column are not touched.

Charging period: The selected row line is connected to Vcc.



CD, DD are the charging and discharging diodes respectively.

Crt is the total capacitance of the reverse biased diodes in the column connected to untouched sensors.

Ct is the capacitance associated with the touching finger, and considered here to be 100pF.

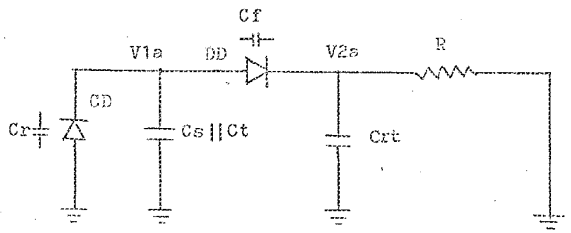
function:

— CD and DD are charged with Qf = Cf * Vd.

— Cs||Ct is charged with Qs = Cs||Ct (Vcc - Vd).

— Crt is charged with Qrt = Crt (Vcc - 2Vd)

Discharging period: The selected row line is grounded.



function:

— 1. CD is discharged by Qf and charged by Qr = Cr * V1a. The charges Qf and Qr come from Cs, and from Crt through DD. Consequently DD is reverse biased, releasing Qf in DD and accumulating a charge Cr * (V2a - V1a).

— 2. For some time later, DD is forward biased. However, if the time interval is very small, not much charge is lost through the timing resistor R and consequently V2a can be easily estimated. This assumption is verified by an experiment using Crt = 68pF, R=100k, Cs=9.4pF and two diodes. The measured time was less than 100 nano-seconds.

The instantaneous voltage V2a for case A can be found as follows.

$$V2a=\frac{Qs+Qrt-Qf-Qr}{Cr+Cs||Ct+Crt}$$

$$=\frac{Cs||Ct*(Vcc-Vd)+0.5*(n-1)*Cr*(V-2*Vd)-Cf*Vd-Cr*V2a}{Cr+Cs||Ct+0.5*(n-1)*Cr}$$

$$V2a=\frac{s*(Vcc-Vd)+0.5*(n-1)*(Vcc-2*Vd)-f*Vd}{s+0.5*(n+3)}$$

where s = Cs||Ct/Cr and f = Cf/Cr.

The time constant for case A, Ta is:

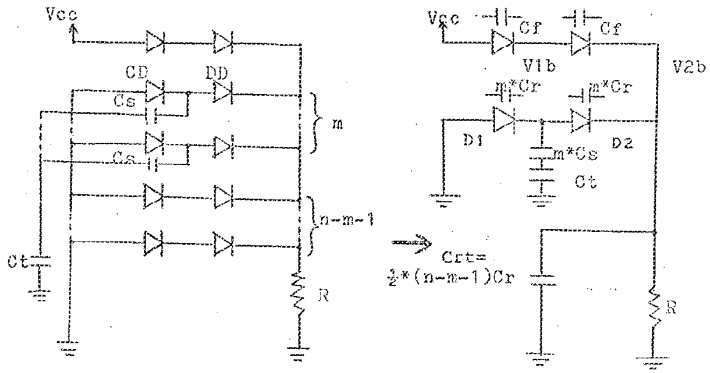
Ta = [Cs + Cr + 0.5*(n-1)*Cr] * R

The discharging voltage (Va) is:

$$Va = V2a * \exp\left(\frac{-t}{Ta}\right) \quad \text{----- equation (1)}$$

Case B: Several m non-selected sensors are touched simultaneously.

Charging period: A selected row line is connected to Vcc.



Cm is the total capacitance of m touched sensor plates with diodes as shown above.

Crt is the total capacitance of reverse biased non-selected, and non-touched diodes.

function:

— CD and DD are charged by QF = Vd * Cf.

— D1 and D2 represent m multiple reverse biased diodes which are charged by Qm = Cm*(Vcc - 2*Vd).

— Crt is charged by Crt*(Vcc - 2*Vd).

Cm is given as follows.

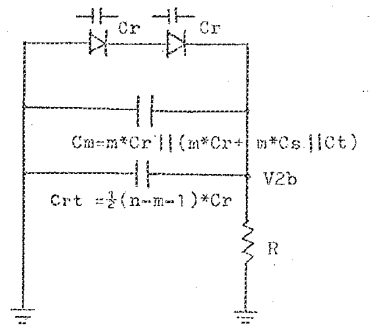
$$Cm=\frac{m*Cr*(m*Cr+m*Cs||Ct)}{m*Cr+m*Cs||Ct}$$

$$\frac{m*(m*s+t)+m*s*t}{2*m*s+2*t+s*t}Cr$$

where s = Cs/Cr, t = Ct/Cr and m is the number of touches.

Crt = 0.5*(n-m-1)*Cr.

Discharging period: A selected row line is grounded.



The instantaneous voltage V2b for this case is:

$$V2b=\frac{Qm+Qrt-Qf}{Qm+Qrt}*\frac{Cm+Crt}{Cm+Crt+0.5*Cr}(Vcc-2*Vd)$$

The time constant for case B, Tb is :

$$Tb = (Cm + Crt + 0.5*Cr) * R$$

The discharging voltage for case B, Vb is:

$$Vb=V2b*exp(\frac{-t}{Tb}) \quad \text{------- equation (2)}$$

Calculation of the separation parameter

From equation 1, the time(Tth) taken for Va to reach the threshold voltage Vt is given by

$$Tth=Ta*ln(\frac{V2a}{Vt})$$

and from the equation 2, the time taken for Vb to reach the threshold voltage Vt is given by

$$Tth=Tb*ln(\frac{V2b}{Vt})$$

Equating the two to solve for m, the separation parameter,

$$Ta*ln(\frac{V2a}{Vt})=Tb*ln(\frac{V2b}{Vt})$$

Using this equation, m can be estimated for various parame-

ter changes. The results are listed below.

TABLE A SEPARATION PARAMETERS

f=Cf/Cr	s=Cs/Cr	t=Ct/Cr	m
12.5	2.5	25.0	8
25	5	25.0	25
12.5	5	25.0	25
25	2.5	25.0	8

s	case 1	case 2	case 3
2.0	6	6	6
2.2	7	7	6
2.4	8	7	7
2.6	9	8	8
2.8	10	9	9
3.0	11	10	10
3.2	12	11	11
3.4	14	12	12
3.6	15	14	13
3.8	17	15	14

case 1: the value of m for f=12.5 and t=20.0
case 2: the value of m for f=12.5 and t=25.0
case 3: the value of m for f=12.5 and t=30.0

APPENDIX B

SUBTRACTION OF OVERFLOWED 8-BIT COUNTER VALUES

This appendix classifies all possible results of subtraction of 8-bit data which are residues of modulus 256. The purpose of this analysis is to demonstrate that when the difference of two data words whose number of bits may be larger than 8, is not greater than 127, it may not be necessary to consider all the bits in the data words. Accordingly in the comparison of the two data residues, only the least significant 8-bits in a word are used and all the other bits are ignored.

Consider the operation A - B. There are 8 cases to consider. Our interest is to find the condition bit settings for an operation on the least significant 8 bits, and compare it with a full word operation.

the case for which A < B using full word comparison

case	(example)	condition bits	
A+	($126) B+ ($128)	S	
A+	($307) B- ($381)	S	
A-	($0B9) B- ($0F3)	S	O
A-	($2F3) B+ ($317)	S	O

S - sign bit is set
O - overflow condition bit is set

the case for which A > B using full word comparison

cases (example)	condition bits
A+ ($369) B+ ($324)	
A+ ($322) B- ($2B3)	0
A- ($3D4) B- ($3C9)	0
A- ($3C7) B+ ($37C)	0

The signs on the letters A and B represent the most significant bit(8-th). If -, then the bit is one, and if + then the bit is zero. The case A- < B+ implies that B+ has undergone one more overflow than A-.

The results show that A > B if the sign is not set, and that A <= B if the sign and zero condition bits are set regardless the condition of the overflow condition bit. This implies that the branch instruction BLE, BGE, BLT, or BGT cannot be used since they all use the overflow condition bit.

APPENDIX C

SUBROUTINES, COMMANDS, AND DATA FORMATS

In this appendix, major subroutines implemented in the dedicated CPU, commands in both local and host modes, and protocols for the communication between the 6809 and the host computer are briefly described.

Subroutines:

NAME: **main**

FUNCTION: Main routine - initializes the tablet and puts it into local mode.

PARAMETER: none

CALLED BY: the 6809 monitor program.

NAME: **wire**

FUNCTION: To transfer terminal input data directly to the host computer.

PARAMETER: none

CALLED BY: main, host; the command in main is w and the command in host is ESC t

NAME: **host**

FUNCTION: Communicates with the host computer. All commands are received from the host computer.

PARAMETER: none

CALLED BY: Wire mode by a command from the terminal input ESC t.

NAME: **bs - binary scan**

FUNCTION: Scans the tablet using the binary scanning algorithm.

PARAMETER: Output routine location; rowaddress, columnaddress, pressure are set before issuing jsr to the output routine.

CALLED BY: main, host

NAME: **ls - linear scanning**

FUNCTION: Scans the tablet sequentially (linearly).

PARAMETER: Output routine location; rowaddress, columnaddress, pressure are set before issuing jsr to the output routine.

CALLED BY: main

NAME: **ms - modified linear scanning**

FUNCTION: Scans the tablet using modified linear algorithm (scans row first and then columns if the row is touched)

PARAMETER: Output routine location: rowaddress, columnaddress, and pressure are set before issuing jsr to the output routine.

CALLED BY: main

NAME: **stream**

FUNCTION: Sends data in stream mode (sends data as long as the tablet is touched)

PARAMETER: none

CALLED BY: host

NAME: **gettemp - get template**

FUNCTION: Gets the codes on a template if there is a template on the top of tablet.

PARAMETER: none

CALLED BY: main, host

NAME: **intp - interpolation**

FUNCTION: Sends rowaddress, columnaddress and a 3 by 3 of pressures for interpolation of touch points.

PARAMETER: none

CALLED BY: main, host

NAME: **init - initialization**

FUNCTION: Initializes the tablet: sets up the reference values and threshold values

PARAMETER none

CALLED BY: main, host

NAME: **average**

FUNCTION: Averages 4 pressure values.

PARAMETER: ctrout(counter output)

CALLED BY: bs, ls, ms

NAME: **readp**

FUNCTION: Reads the pressure of a group of sensors

PARAMETER: rowvec(the row vector that is, the bit pattern for the row selection registers), colvec(the

column vector that is, the bit pattern for the column switch selection register)

CALLED BY: bs, ls ,ms

Commands in local mode

t - shows threshold values for all grouping levels

r - shows reference values for a grouping level

l - linear scanning

b - binary scanning

m - modified linear scanning

a - manual threshold input

c - change the grouping level for the linear scanning mode

i - initialize the tablet (set up the reference and threshold values)

n - get a set of new thresholds

Commands for the host computer

g - go; input from the 6809 (tablet ready signal),
- the host can send a command upon receiving g

i - initialization; output to the 6809

-initialize the tablet; gets new thresholds and reference values

n - set new thresholds; output to 6809

w - send the thresholds for all levels

r - send the reference values to the host

d - delta mode (send data only if there is a delta change)

- send a pixel only if there is a change

(not implemented)

s - send the data in stream mode

t - send the pattern of the template lying on the top of the tablet

x - go back to wire mode so that terminal can talk directly to the host

Data formats

s scan mode: stream mode

r pixel pixel .....ff r pixel ....

where pixel = xxyyzz

— xx, yy, and zz are row, column, and pressure in hexadecimal ascii values.

p scan mode: interpolation mode

r ipixel ipixel ... ff r ipixel .....

where ipixel = xxyyzzzzzzzzzzzzzzzz

— the position of the adjacent pixels is (-1,-1)(0,-1)(1,-1)(1,0) (1,1)(0,1)(-1,0)(-1,1)

w mode: threshold values for all levels

(t(i,j),i=0,7),j=0,6)

where i is the column level, j is the row level and t is an array of two byte hexadecimals.

t mode: template pattern

r pixel pixel .... ff

where the sequence of pixels is the template pattern.

r mode: reference data format

— specify the sensor grouping level(rl, cl) to the 6809

(z(coladr,rowadr),coladr = 0, lca), rowadr = 0, lra)<CR>;

where lca = 1 << cl - 1 and lra = 1 << rl -1.

General communication protocol

originated in 6809 originated in the host

g ----------> ready to send a command

<--------- command

echo ---------> ready to receive the data according to the command

-

all the data formats shown above are inserted here

-

goback to start point

New command input is executed between the end of a scan mark ff and the begining of a scan mark r in the protocol.

W. Daniel Hillis
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

A High-Resolution Imaging Touch Sensor

Abstract

A dexterous robot manipulator must be able to feel what it is doing. The mechanical hand of the future will be able to roll a screw between its fingers and sense, by touch, which end is which. This paper describes a step toward such a manipulator, an imaging tactile sensor with hundreds of pressure sensors in a space the size of a fingertip. The sensor was designed as part of a tendon-actuated mechanical finger, similar in size and range of motion to a human index finger (Hillis 1981). As a demonstration, the device was programmed to distinguish among several commonly used fastening devices—nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws—by touching them with the finger and analyzing the tactile image.

This paper describes how the tactile sensor array was constructed and how it works. The simple pattern-recognition techniques used in the demonstration program are also outlined. The final section notes some promising directions for future research.

The Sensor

The touch sensor is a monolithic array of 256 tactile sensors that fits (appropriately) on the tip of a finger. This resolution is comparable to that of the human forefinger. Each sensor has an area of <0.01 cm^2 and gives an independent analog indication of the force over its surface over a range of 1–100 g. The array is scanned one column at a time to minimize the number of connecting wires. The sensor is rugged, flexible, and has a skinlike texture.

The touch array has two conductive components: a flexible, printed circuit board and a sheet of anisotropically conductive silicone rubber (ACS). The ACS has the peculiar property of being electrically conductive along only one axis in the plane of the sheet. The printed circuit board is etched into fine parallel lines, so it too conducts in only one dimension. The two components are placed into contact with the lines on the printed circuit board perpendicular to the ACS axis of conduction. The contact points at each intersection of the perpendicular conductors form the pressure sensors.

The device also includes a separator to pull the conducting layers apart when pressure is released. The sensitivity and range of the sensor depend largely on the construction of this intervening layer. There is a trade-off between sensitivity and range. For a large pressure range, the best separator tested was the woven mesh of a nylon stocking. For high sensitivity, a separator may be deposited directly onto ACS by spraying it with a fine mist of nonconductive paint. The conductive rubber presses through the separator so that the area of contact, and hence the contact resistance, varies with the applied pressure.

The pressure/resistance relationship is nonlinear, as shown in Fig. 5. We do not have a model of the contact mechanism that quantitatively explains the change in resistance with applied pressure; however, Fig. 1 illustrates a plausible qualitative model. Pressure on the elastomeric ACS deforms the material around the separator, allowing it to contact the metal below. Larger pressures result in more deformation and larger contact areas. If the resistance of the contact is proportional to the contact area, the contact resistance will be inversely related to the applied pressure. For the object-recognition application, the nonlinear response of the sensor was not a significant drawback.

This report describes research done at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. The research was supported in part by IBM, under an agreement with the Massachusetts Institute of Technology Laboratory for Computer Science. Support for the Artificial Intelligence Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under contract with the Office of Naval Research, contract N00014-80-C-0505. The author is supported by a fellowship provided by the Fannie and John Hertz Foundation.

The International Journal of Robotics
Research, Vol. 1, No. 2, Summer 1982,
0278-3649/82/020033-12 $05.00/00

Fig. 1. Contact resistance changes with changing area.

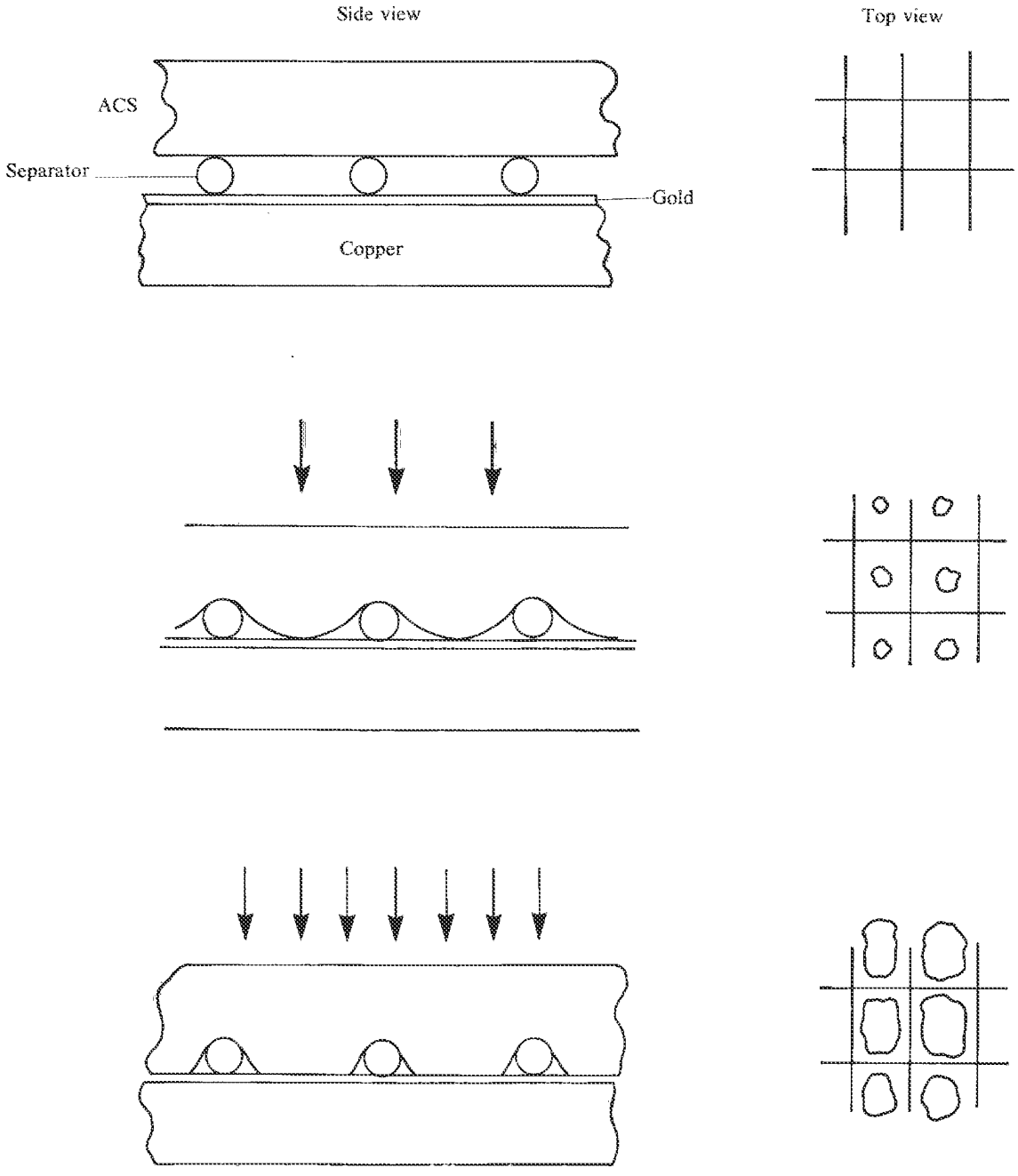
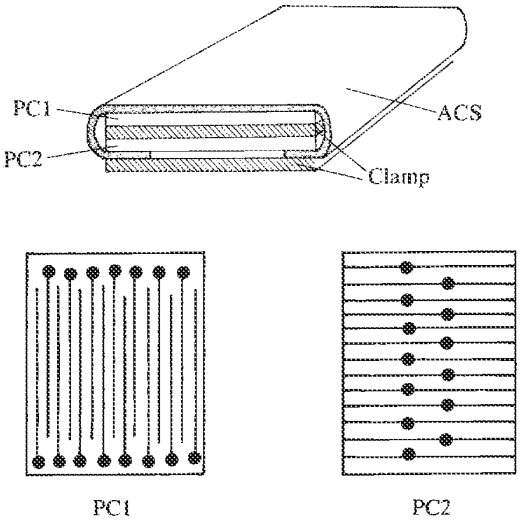


Fig. 2. Mechanical drawing of touch sensor. ACS = anisotropically conductive silicone rubber; PC = printed circuit board.



The ACS itself is constructed of layers of silicone rubber impregnated with either graphite or silver, alternating with similar nonconductive layers. Each layer is approximately 250 μ thick. The layers are oriented at right angles to the plane of the sheet. The linear resistivity, in the conducting direction, is on the order of kilohms per centimeter for graphite-impregnated ACS. This is inconveniently high for building large sensors, and we were able to lower it to approximately 100 Ω per centimeter by electroplating the sheet with gold. It is possible to plate only over the conductive silicone, so that the cross-resistance remains essentially infinite. Silver-impregnated silicone rubber has a substantially lower bulk resistance, but we were unable to obtain the material in the proper form. The minimum resolution of commercially available ACS is about 50 lines per centimeter.

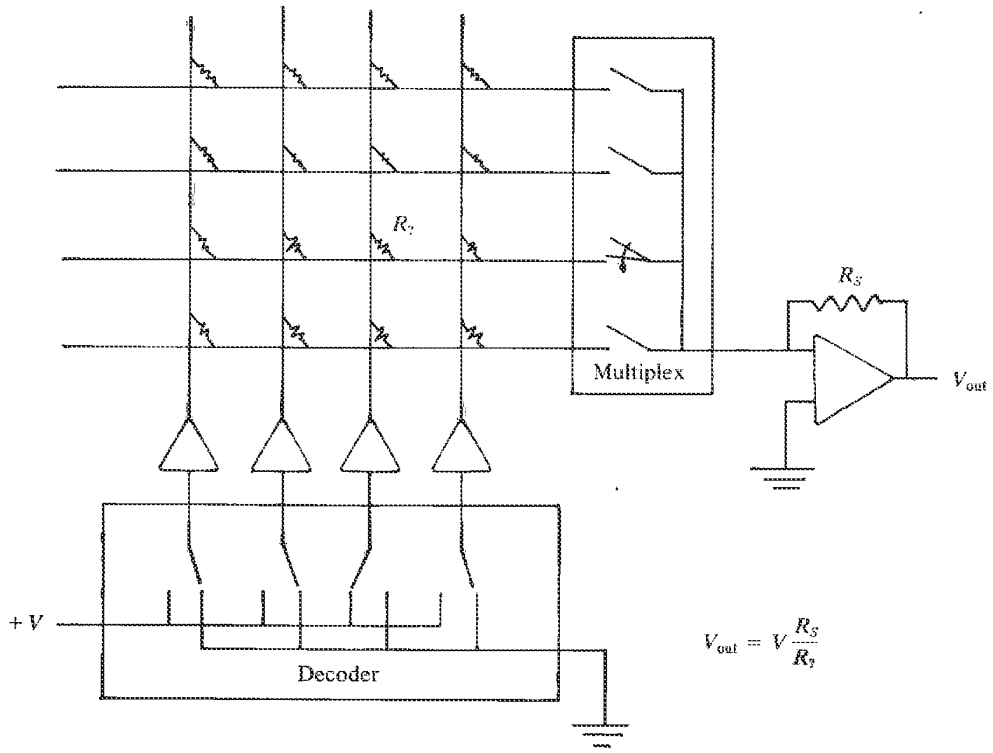
Wires are soldered to the edges of the printed circuit board. The ACS is mounted so that its edges fold around the printed circuit, where they are pressed against contact fingers on the other side (see Fig. 2). A compound sensor with high range and good sensitivity may be constructed by placing a high-range (nylon mesh) sensor behind a sensitive one. In this case, the flexible circuit board in the front layer was eliminated, so that the center layer of ACS was shared between the two arrays.

Scanning the Array

Attaching wires only at the edges of the array reduces the number of necessary connections. This is important given the limited space of a mechanical finger. (The 256-cell sensor used 32 wires, #42, stranded. The resulting cable is <3 mm in diameter.) The array is scanned by applying a voltage to one column at a time and measuring the current flowing in each row. A potential problem with this method is the introduction of "phantom" tactile images. When multiple points are activated simultaneously, it may appear that untouched points are also conducting. This is the analog version of the crosspoint problem in *xy*-scanned keyboards: if three out of four switches on a rectangle are closed, the fourth appears to be also. This happens because the path through the other three connections is electrically in parallel with the phantom connection. In keyboards, it is usually avoided by putting a diode at each point of intersection. This could be done for touch array also, but it would add considerably to the complexity of the device and it might also introduce undesirable mechanical stiffness. With resistive contacts, it is theoretically possible to compute the actual resistances from the measured resistances by solving N equations in N unknowns (Larcombe 1976), but the technique tends to amplify errors due to inaccuracy of measurement, noise, and resistance along the conductive axis. Other researchers (Stojiljkovic and Clot 1977; Broit 1979; Harmon 1982) have avoided the problem by attaching a separate wire to each sense point. This is impractical for high-resolution arrays and, again, it limits mechanical flexibility.

Instead, we used the scheme illustrated in Fig. 3. It is similar to the voltage-mirror approach suggested by Purbrick (1981). A fixed voltage is placed on the column of interest, while all other columns are held at ground potential to ground out any alternate paths. The rows are all held to ground potential also, by injecting whatever current is necessary to cancel the current injected by the active column. The value of the resistance of a crosspoint is inversely propor-

Fig. 3. Scanning the array.



tional to the current that is necessary to pull the corresponding row to ground potential. By this method, extraneous columns are at the same potential as the rows (ground), so no current will flow through the unmeasured crosspoints. The holding currents depend only on the column drive voltages and the resistances in question. The entire array is scanned by measuring one column at a time, as described above.

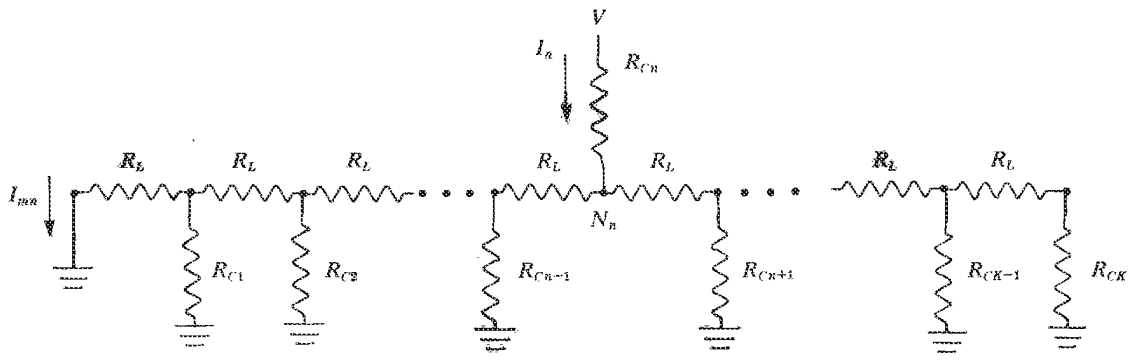
The method described is valid only if the crosspoint resistances are high compared to the linear resistances of the row and column lines. Otherwise, it is not possible to hold an entire row or column at a fixed potential. This effect may be understood by referring to the electrical model of a single row illustrated in Fig. 4. The applied voltage (V) and the linear resistance (R_L) are known, but the unknown resistance (R_{Cn}) cannot be determined unless the potential at node N_n is known. This potential may be computed, in time proportional to the number of nodes, by first measuring the unknown reistances near the edge. The resistance of all the the unknown contacts may be determined by computing the successive two-port parameters of the subnetworks toward the edge of the unknown node.

$$Z_0 = 0$$
$$G_0 = 1$$
$$R_n = \frac{V - I_{mn}G_{n-1}(Z_{n-1} + R_L)}{I_n}$$
$$Z_n = \frac{R_n(Z_{n-1} + R_L)}{Z_{n-1} + R_n + R_L},$$

where V is the applied voltage, I_n and I_{mn} the measured currents, Z_n the input impedance, and G_n the voltage transfer ratio of the network to the left of R_n.

For large arrays it may actually be necessary to compute the resistances as shown, but as long as the linear resistance is low compared with the contact

Fig. 4. Electrical model of one row.



resistance, this is not necessary. If the measured values are used directly, then the worst-case error for a row of N elements is

$$\frac{R_{\text{measured}}}{R_{\text{actual}}} = \frac{R_n + NR_L}{R_n}.$$

This is easy to determine because the worst-case occurs when all contacts, except for the one being measured, are open. Other contact closures will only lower the potential of node N_n, increasing the accuracy of the measurement. The error may also be reduced by a factor of two by making contact at both ends of the row.

Performance

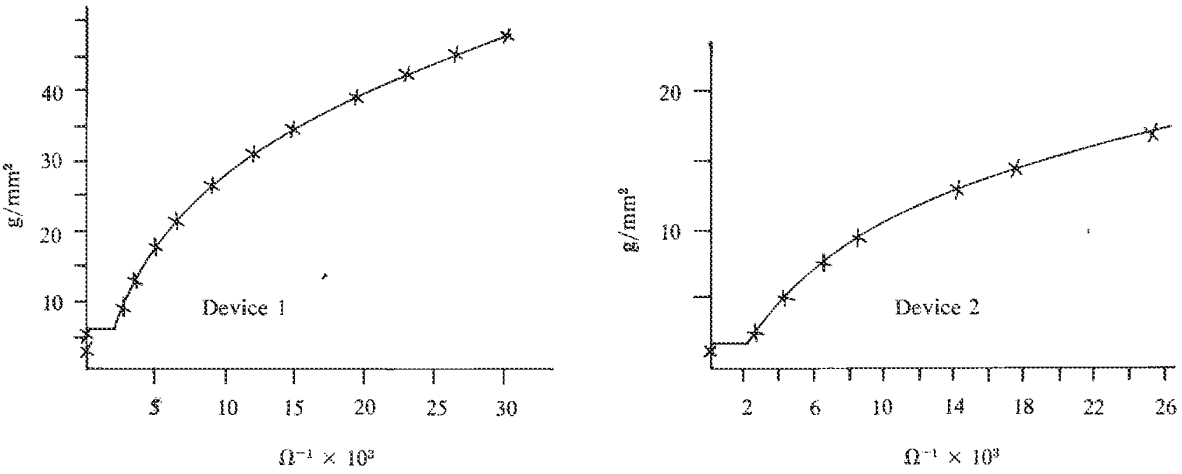
Several sensory arrays were contructed with varying range and resolution. Figure 5 shows pressure-resistance curves for two representative devices. Device 1 has a sprayed separator (approximately 10^4 dots per square centimeter). The separator of device 2 is nylon mesh (Leggs, Extra Sheer). The ACS used in device 1 was plated with gold on the contact side. All devices showed good mechanical durability and, after an initial settling period, stable electrical characteristics. (The first prototype, almost a year old, shows no noticeable change in contact resistance.) The highest-resolution device (1) was a 16-by-16 array, 1 cm in area. This is the sensor used with the finger. Sample images of the top of a screw, an electronic connector, a ⅛-in ring, and a cotter pin are shown in Fig. 6.

Method of Use

The touch sensor was mounted on a tendon-controlled finger that has approximately the same shape, size, and range of motion as the human forefinger. Like the human finger, it has three joints: a pivot with two degrees of freedom at the base and two hinge joints with one degree of freedom. These joints are controlled by four pairs of tendons, which are driven from four electric motors mounted behind the base of the finger. Torque and position are measured only at the motors, so that joint torques and angles were computed as described below. The finger and associated motors are mounted on a fixed base, with a small platform, extending just below the finger, on which objects may be placed for testing.

The tendons of the finger are arranged in opposing pairs—one bends the joint, the other straightens it. A system of pulleys keeps the total length of each pair constant. This allows both ends of a tendon pair to be driven from a single motor. The lever arm of the tendon pulling against the joint is kept constant over all angles by winding the tendon over a pulley fixed to the joint. The tendons in the mechanical finger are arranged to give independent control of the torque and angle of each joint. The finger and its control are described in more detail elsewhere (Hillis 1981).

Fig. 5. Performance curves of two sensors.



The tendon finger and the touch sensor were used together in the discrimination program. The program used the finger to press and probe the object placed in front of it. Based on how the object felt, the program guessed the shape and orientation of the object. The device was programmed to recognize commonly used fastening devices such as nuts, bolts, flat washers, lock washers, dowel pins, cotter pins, and set screws. The program was written in Lisp and ran on a specially augmented Lisp Machine (Weinreb and Moon 1979), with independent microprocessors to control low-level input/output functions.

The program worked, as will be described, but it never worked well. Its most impressive weakness was that it would confidently identify any object placed in front of it as one of the six test objects. It is, however, a starting point. Many of the techniques used in the discrimination task should be applicable in more sophisticated programs of the future.

Why Touch Is Easier than Vision

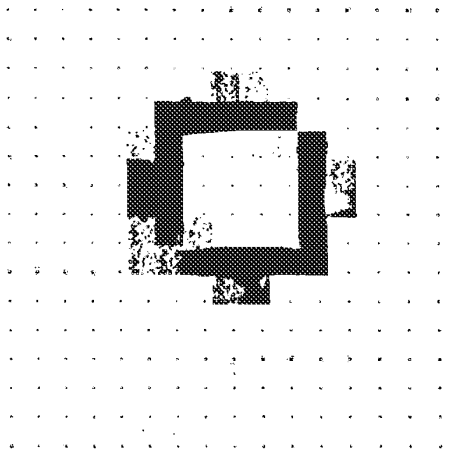
In writing the discrimination program, we shamelessly retraced the steps of early researchers in machine vision. There is good reason to believe that these simple techniques have a better chance of working in the tactile domain than they did in the visual. For one thing, there are far fewer data to be analyzed than in a visual image. This means that even with a high-resolution tactile array, complex processing may be performed in real time. Another factor is that collection is more readily controlled. Since placement and pressure of the fingertip are controlled by the program, analyzing a tactile image is like analyzing a visual image with controlled background, illumination, and point of view.

There is also a third factor responsible for making tactile recognition the easier task: the properties that we actually measure are very close, in kind, to the properties that we wish to infer. In vision, it is only possible to discover mechanical properties (shape, orientation, absolute position) by deducing them from optical properties (shading, projection, reflectivity). In touch, we measure mechanical properties directly.

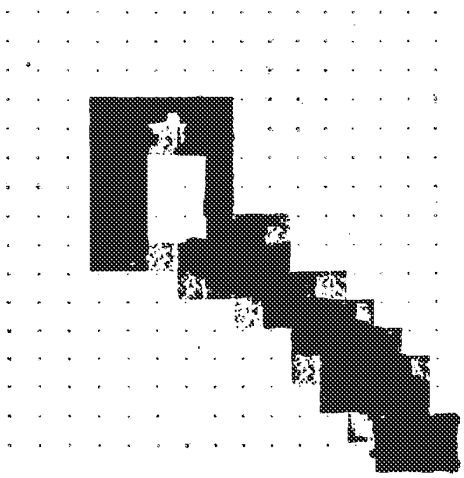
Active Sensing

There are two possible approaches to any kind of sensory recognition. The first is the *analytic approach*. In the analytic approach, we start with an image of some sort, extract features of the image, and then, from the features, abstract some kind of a

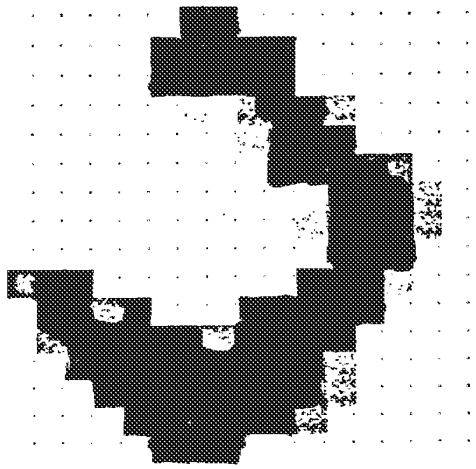
Fig. 6. Sample tactile images from the sensor. Approximate actual sizes of objects shown below images.



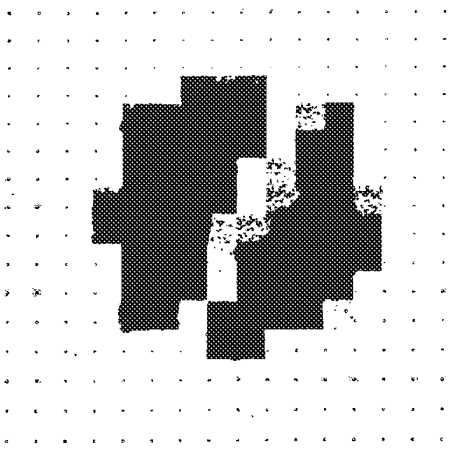
⅛-in. ring



Cotter pin

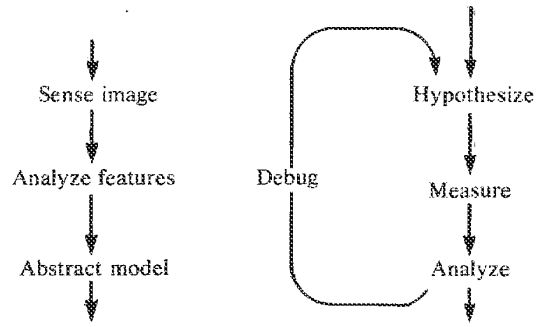


Spade lug



Top of screw with slot

Fig. 7. Two approaches to recognition.



model of what is shown in the image. The analytic approach is bottom-up, or data-driven. The second approach, the *active approach*, is top-down, or knowledge-driven. When taking the active approach, we begin with a hypothesis of what is in the image. Based on that hypothesis, we make measurements or perform experiments to test the validity of the hypothesis. The results of the theory are then analyzed using the same techniques as the analytic approach, and, based on the result of the analysis, the hypothesis may be modified or confirmed. If it is modified, we try experiments to test the new hypothesis, and so on, until we arrive at a conclusion that agrees with the data. The two approaches are represented schematically in Fig. 7.

We believe that the active approach is more appropriate in the tactile domain. For one thing, the information in a single image is often insufficient for recognition of the object. Moving the finger to make different measurements is the best way of collecting enough data, and in order to decide how to move the finger, the program must have some expectation of what object it is feeling. The program should operate at all times with a hypothesis of what it is feeling. It recognizes the object by actively probing it with the finger, modifying its internal "hallucination" of the object to conform with the measured reality.

The Domain: Nuts and Bolts

For the purpose of testing the sensor, the finger, and the recognition techniques, we chose a restricted range of test objects that the program would be expected to recognize. Specifically, we chose a set of commonly used mechanical fastening devices that are important for potential industrial applications of robotics. Recognition of fasteners and determination of their position and orientation when they are grasped by a manipulator is an important industrial problem. It is also a problem that is unlikely to be solved by machine vision because the hand obscures the object and because forces cannot be seen. In this domain, tactile and visual sensing would complement each other well: vision for locating objects and measuring their absolute position, touch for sensing local shape, orientation, and forces once they are grasped.

The particular objects, chosen because they have simple shapes that are easy to represent and easy to distinguish, were machine screws, set screws, flat washers, lock washers, dowel pins, and cotter pins. Restricting the range of possible objects to such a small and easily distinguishable subset makes the recognition task less difficult, but it also introduces the possiblity that the recognition methods used are not really generally applicable. This may be the case here. Recognizing a larger range of objects would certainly require identifying a richer set of tactile features.

The objects were all small (not more than ½ in. in any dimension). In general, the smallest commonly used size in a given category was used for testing. For example, the machine screw was #0-80 by ⅜ in. and the cotter pin was ½ in. long by $^1/_{16}$ in. in diameter. Using such small objects allowed the entire image to be read in one impression of the sensor. This avoided the problem of coordinating multiple sensor impressions into a single tactile image.

Representation: Describing How Something Feels

Based on features of the test object that make it identifiable by human touch, a simple language was developed for describing the object's tactile properties. Three categories of features, or parameters, were chosen to represent the feel of an object.

Fig. 8. Table of objects.
+ = bump; 0 = depression.

1. Shape. For the fastener micro-world there are only two possible shapes: round and long. The shape may therefore be determined directly from the height/width ratio of the image.
2. Bumps. The locations of local pressure anomalies are expressed in object-relative coordinates, for example, in coordinates relative to the major and minor axes of the image. Bumps may be positive (convex) or negative (concave). (In the implemented program, only the sign and position of a bump are considered significant for matching purposes. There is no intensity information.)
3. Stability. This property indicates how easy it is to roll the object in various directions. (In the program, two Boolean values represent the stability of an object. These indicate whether or not the object rolls freely along each of the primary axes.)

These three parameters were sufficient for distinguishing among any of the objects in the test set. Figure 8 shows which properties were exhibited by each object.

		Shape	Bumps	Stability
	Machine screw	Long	+	Yes
	Dowel pin	Long	0	Yes
	Cotter pin	Long	0	No
	Set screw	Round	0	Yes
	Lock washer	Round	+	No
	Flat washer	Round	0	No

Implementation

The control tasks are divided among a Lisp Machine and five Z-80 microprocessors. Most of the code is written in Lisp and ran on the Lisp Machine. Each microprocessor is dedicated to a simple task with real-time constraints. One scans the tactile image array, while others take specifications for joint motions and execute them in real time. The memories of the microprocessors are directly accessible by the Lisp Machine. This shared memory provides a convenient interface between the two levels of processing. The output of the tactile array, for example, is available as a Lisp array object.

The top-level loop program is just a translation of the active approach diagrammed in Fig. 7. The program begins by assuming that the object is whatever best matches the information that it has so far. This is the "hypothesize" step. It then tests assumptions that it has made about the object and, unless they are all correct, repeats the process from the beginning. Testing the assumption may involve moving the finger to roll or probe the object, or may just involve making a computation from already collected data.

Flow of control in this process depends on what information is needed. When a property queried is not already known, it is computed or measured. In the process of computing the value, the program may make queries about properties that, in turn, may need to be computed. For example, if we ask an object's shape and it is not known, then it must be computed from the object's dimensions along the primary axes. If these axes are not known, they must be computed from the image of the object. If no image has been read in, the finger must be pressed against the object, and so on. This is "call by need" control flow. It prevents information from being measured or computed unless it is actually needed in the recognition process.

After an image is read in, it must be processed to determine the object's location, the primary axes, and the location of any bumps. The first step in processing the tactile image is the elimination of unwanted detail. This is accomplished by averaging the value of each pixel with those of its immediate neighbors. The image is then contrast-enhanced to two bits per pixel by comparing it to fixed threshold values. If the offset pressure has been chosen properly (see below) the four possible pressure values correspond to background, depressions, primary figure, and bumps.

Next, the aspect ratio and major and minor axes of object are determined. Computation of the aspect