

SELF-TRACKER:
A Smart Optical Sensor on Silicon

by

Gary Bishop

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

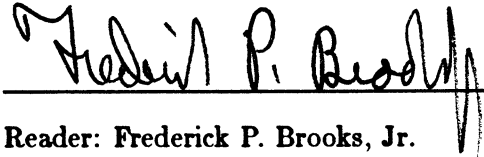
Chapel Hill

1984

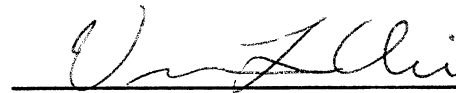
Approved by:



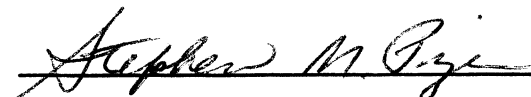
Advisor: Henry Fuchs



Reader: Frederick P. Brooks, Jr.



Reader: Vernon L. Chi



Reader: Stephen M. Pizer

Copyright © 1984
Gary Bishop

Acknowledgements

Thanks to

Xerox Corporation for fabricating several of my chips.

Vern Chi, Mark Monger, John Poulton, and John Thomas of the UNC Computer Science Microelectronic Systems Laboratory for friendly and helpful hardware support.

Guerry Waters and Savannah Electric and Power Company for financial support.

John Poulton for revealing many of the mysteries of MOS circuits.

Fred Brooks, Vern Chi, Henry Fuchs, Steve Pizer, and Jim Stasheff for serving on my committee.

Vern Chi for the kernel of the outward view idea and for his many helpful insights.

Henry Fuchs for the inspiration provided by his energy and love for computing and for directing my research by asking questions, pointing out problems and suggesting experiments.

Fred Brooks for the opportunity to learn from him while I served as Executive Officer and for the inspiration provided by his personal and professional life.

John Zimmerman for his insight into the meaning and conduct of science, and for his friendship.

Mother for her love, and for many trips to the public library.

Jonah for bringing immeasurable joy to my life.

Ivy for her love, and joy and for believing that I could do it.

Table Of Contents

1	Introduction	1
1.1	The Problem	1
1.2	The Inspiration	2
1.3	A Proposed Solution	2
1.4	Design Issues	3
1.5	Overview	4
2	Outward-view Tracking	5
2.1	Review of Previous Work	5
2.2	New Tracking Method	7
2.3	Alternative Implementations	8
2.3.1	Tracking with Beacons Only	9
2.3.2	Tracking in a Natural Environment	12
2.3.3	A Combined System	13
2.4	Choice of a Method for Further Study	14
3	A Natural-Environment Tracker	15
3.1	Design of the Cluster	15
3.1.1	Goals for the Cluster Design	15
3.1.2	Design Equations for the Cluster	16
3.1.3	Proposed Cluster Designs	20
3.2	Design of the Sensor Chip	21
3.2.1	Frame Rate and Simplicity	22
3.2.2	Imaging System	24
3.2.3	Registration System	28
3.2.4	Communication	30
3.3	Design of the Motion Extraction Algorithm	31
3.3.1	Problem Formulation	31
3.3.2	Solution Methods	32
3.4	A Simulation Study of Accumulated Error	33
4	A Chip for Natural-Environment Tracking	35
4.1	Motivations for Implementation	35
4.2	Description of Chip	35
4.2.1	Imager	36
4.2.2	Processor	39
4.2.3	Control and Signal Generation	45
4.3	Testing Environment	49
5	Photosensor Design	51
5.1	The Optical Mouse Photosensor	51
5.2	An Improved Photosensor	52
5.2.1	Sensitivity of the Improved Photosensor	54
6	Next Steps	56
6.1	Task List for a Natural Environment Tracker	56
6.1.1	Description of Tasks	57
6.2	Further Research for a Combined System	59
	References	60

Chapter 1

Introduction

1.1 The Problem

In the last 20 years much progress has been made toward realistic computer display of three-dimensional objects. It is now common to display pictures with hidden surfaces, smooth shading, and realistic lighting. Yet when we try to interact with these realistic three-dimensional images we are forced to use devices similar to the remote manipulators used while handling radioactive materials. Ivan Sutherland, in 1965 [Sutherland, 1965], suggested that a solution to this problem was to have a room within which the computer could control the existence of matter; then users could experience and interact with computer-generated objects just as they do with physical objects. In 1968 he described the head-mounted display [Sutherland, 1968; Vickers, 1974; Clark, 1976], an approximation to the ideal display that generated realistic images in the user's space. The fundamental idea of a head-mounted display is to present the user with a perspective image that changes as he moves. The image is presented using small CRT's mounted on a helmet and it is updated in real time based on the position and orientation of the user's head so that the computer-generated objects appear to be in the room with the user.

The two major components of a head-mounted display are image generation and tracking. The image generation system of Sutherland's head-mounted display was successful; its descendants are in wide use today as real-time line-drawing systems and raster flight simulators from Evans and Sutherland Computer Corporation. The three-dimensional tracking problem, however, was never satisfactorily solved. Although numerous methods were tried then and have been developed since, the dominant computer input devices are still two-dimensional and no three-dimensional input system has gained wide acceptance.

Three-dimensional tracking for a system such as a head-mounted display is difficult because it must be fast and accurate over a large volume, with little restriction of the user or the environment and must determine both position and orientation. As evidenced by previous work on this problem (see section 2.1), this combination of characteristics is difficult to achieve in a single system.

Although the motivating application for a system of this kind is in a head-mounted display, it could also find application in conventional graphics systems as an unconstrained three-dimensional input device, in interactive surface design, in generation of descriptions of objects for computer display, and in human and animal gait and motion studies.

1.2 The Inspiration

The inspiration for this research came from Vernon Chi of the UNC Computer Science Microelectronic Systems Laboratory, who suggested that we could track in three dimensions using something similar to Richard Lyon's Optical Mouse, imaging the room rather than special paper. The Optical Mouse [Lyon, 1981] is a pointing device for positioning the cursor on a display. The mouse is moved around on a pad to move the cursor on the display. Unlike earlier electro-mechanical mice, Lyon's circuit detects motion optically using a single downward-looking integrated circuit, light source, lens, and paper with a special dot pattern. It uses the light-sensitive properties of nMOS integrated circuits and a "mostly digital" circuit to produce binary snapshots of the dot pattern. It tracks the features in these binary images using an inhibition network matched to the pattern. The entire system, optical sensor, memory, processing and computer interface is realized on a 3.5 by 4.5 millimeter nMOS circuit with 5 μm features.

1.3 A Proposed Solution

The result of this research is SELF-TRACKER, a new method for real-time computer tracking that uses a cluster of outward-looking custom integrated circuits as smart optical sensors. Each custom integrated circuit measures and reports the shifts in its one-dimensional image of the stationary room environment. These shifts will be processed by a separate general-purpose computer to extract the three-dimensional motion of the cluster.

The fundamental concept of the sensor chips relies on a cyclic relationship between speed and simplicity. If the frame rate is fast, consecutive images will be only a little different. Small image changes can be tracked with a simple algorithm. This simple algorithm can be implemented with small circuitry. The small circuitry lets a single chip hold the complete sensor, both imaging and image processing. Such implementation allows each sensor to be fast because all high-bandwidth communication is done on-chip. The small size also allows many independent sensors to be placed into the small sensor cluster. This cyclic relationship can spiral up as the design is iterated, with faster and simpler operation, or down, with slower and more complex operation. The system design sequence described here has been spiraling up.

1.4 Design Issues

The following table is a brief outline of my approach to the major issues in the design of SELF-TRACKER.

Issue	Attack
What 3-D tracking method to use for a head-mounted display?	After building a prototype TV camera tracking system that could achieve only three updates per second, I studied other tracking methods and concluded that an outward-view system could offer greater portability, larger working volumes, and a less restricted environment than other systems.
What will the outward-view system see?	Mathematical analyses and simulations were done for three different outward-view tracking schemes: one sighting beacons in the environment, one integrating motions in the natural environment, and a combined system that uses both beacon and motion sensors. The combined system is most attractive.
How rapidly do errors accumulate?	The natural-environment tracker measures motions, not absolute position and orientation, so it can drift. A computer simulation of the natural-environment tracker shows that a natural-environment system could operate for several seconds without excessive error accumulation, during which time one or more beacons will come into view.
How can the 3-D motion of the cluster be determined?	I investigated several non-linear solution methods and developed a linear (approximate) method that is fast and accurate.
How fast must a motion sensor operate to assure small image changes?	Time and motion literature, and measurements of human motions using digitized images from a television camera showed that 1000 frames per second captured natural motions and that 3000 frames per second could handle the fastest motions that I measured.
Can nMOS photosensors be made sufficiently sensitive?	Four test chips of two different sensor designs yielded photosensors that are small and reliable, and operate at 1000 to 5000 frames per second in ordinary room light with a small lens.
Should 1-D or 2-D images be used?	A 1-D photosensor array can be designed on the 2-D chip surface with much closer pixel spacing than a 2-D array. To determine if 1-D images of natural scenes contain enough trackable features, I simulated the operation of a 1-D photosensor array using digitized images from a standard television camera. 1-D images work very well for small image shifts.
How should the images be represented and processed?	Experiments with a variety of bi-level image representations using the simulated 1-D images showed that the arithmetic sign of the slope of the intensity captures the essential image features (the edges) and is simple to generate.
How can the image shift be measured?	Small image shifts can be measured accurately by finding that relative shift between two sign-of-slope images which minimizes the bit-wise disparity.

How can the minimum disparity be found?	Simply counting the 1's in the output of the XOR gates and comparing the counts is not possible because of time, area, and accuracy constraints. Instead, the string of 1's and 0's from XOR gates is converted into a unary representation by asynchronously packing all the 1's together at the far right. This unary representation is simple and fast to generate and compare.
Is a single-chip sensor possible with available technology?	I have designed, fabricated through MOSIS, and tested a prototype motion sensor chip. The chip is $6800 \times 6300 \mu\text{m}$ with $4 \mu\text{m}$ features.

1.5 Overview

SELF-TRACKER is introduced and analyzed in chapter 2. The functions and performance requirements of the custom integrated circuits that compose the natural-environment SELF-TRACKER cluster and of the general-purpose computer that does three-dimensional analysis are the subject of chapter 3. Chapter 3 ends with a simulation study of the accuracy of the natural-environment SELF-TRACKER.

A chip that includes most of the function described in chapter 3 has been implemented and tested as a partial demonstration of the feasibility of this new approach. Details of its design and testing are given in chapter 4.

High-speed operation in ordinary room light with imaging and image processing on the same chip requires a photosensor design that is sensitive and that can be fabricated using standard nMOS fabrication processes. A new photosensor circuit, suggested to me by Carlo Séquin, has been fabricated, tested, and shown to be sensitive, small, consistent, and reliable. Chapter 5 includes the circuitry, theory of operation, layout, and test data for this photosensor as well as the photosensor in Lyon's Optical Mouse.

Chapter 6 outlines the steps necessary to achieve a complete three-dimensional computer input system and suggests some related areas of research.

Chapter 2

Outward-view Tracking

This chapter describes a new method for three-dimensional tracking that uses a small cluster of outward-looking sensors. After a short review of previous tracking systems, the new method is described in general terms and three different implementations are outlined. The implementations are described in this chapter with only enough detail to show the design parameters and possible problems with the implementation. One implementation, the natural-environment SELF-TRACKER, is described more fully in chapters 3, 4, and 5.

2.1 Review of Previous Work

Commercial and experimental three-dimensional computer input systems have been based on acoustic, magnetic, mechanical, and optical sensing. Commercial and commonly used systems are listed here before experimental systems.

Acoustic systems, such as the commercial Spacepen [Science Accessories, 1970], track three-dimensional position using a movable spark-gap source and fixed ultrasonic microphones. Time-of-flight ranging, based on the speed of sound in air, is the measurement method. It works well for small working volumes but suffers accuracy problems caused by variations in air density and by air motion when the working volume is enlarged. It does not sense orientation.

The Polhemus cube [Raab *et al.*, 1979], from Polhemus Navigation Sciences, determines three-dimensional position and orientation using orthogonal magnetic fields and a small, precision magnetic coil assembly. The commercially available system provides a limited working volume, a 1-meter radius sphere, but a specially modified system provides much greater range for the Air Force research group at Wright-Patterson Air Force Base. The calibration of these systems is affected by the presence of conducting materials in the working volume, but once proper calibration is achieved, they are accurate. Polhemus also has the advantage that the magnetic pickup that is attached to the user is small, lightweight, and passive. The commercial system costs only about \$20,000 but the expanded-range system used by the Air Force is much more expensive.

Mechanical linkage systems such as the custom system used at the University of Utah [Vickers, 1974], the remote-manipulator arm at UNC [Kilpatrick, 1976], and the Noll box [Noll, 1971], have limited range, about 1.5 meters in any direction, and are restrictive due to the mechanical connection to the user and to friction, backlash, and inertia in the mechanical system. Also, they make it very difficult to track multiple objects (for example the head and the hand).

All the remaining systems are optical and are listed with commercial systems first followed by experimental systems in order of their publication.

SELSPOT [Woltring, 1974] is a commercial system marketed from Sweden by Selective Electric Corporation. It uses camera-like, fixed sensors that rely on the sensitivity of a lateral-effect photodiode [Wallmark, 1957] to the position of a light spot on its surface to determine X-Y location. A pair of these cameras can determine three-dimensional position in a 1-cubic-meter working volume, using well-known stereometric techniques. The system can track up to 30 light emitting diodes (LED's) at 315 samples per second. Its working volume can be extended using three or more cameras but they are expensive (about \$40,000 for a system with 2 cameras plus about \$12,000 for each additional camera). It cannot directly determine orientation but can infer it if the target lights have a known and fixed spatial relationship.

OP-EYE [United Detector Technology, 1981], another commercial lateral-effect photodiode system is similar to SELSPOT but is intended for the microcomputer market (about \$4000). It can track a single light source with an advertised resolution of 1 part in 4000 at 5000 samples per second. Like SELSPOT, it provides limited working volume and cannot directly determine orientation.

Twinkle Box, an experimental system developed by Burton at the University of Utah [Burton, 1973], used four fixed sensors to detect light from sequentially blinked LED's that were attached to the user. Each sensor consisted of a slotted disk rotating at 3500 revolutions per minute in front of a pair of photomultiplier tubes. With a sensor at each corner of the room, the system provided a working volume of 112 cubic meters with mean error of 7.3 millimeters but it required a special flat-black room and generated considerable noise because of its rapidly rotating slotted disks. It could track up to 61 lights per second in a real-time mode or 925 per second in an off-line (record then calculate) mode, but like the systems described above, it could determine orientation only from multiple lights with fixed spatial relationship.

The experimental CCD based system developed by Fuchs, Duran, and Johnson [Fuchs *et al.*, 1977] used fixed sensors that consisted of a knife edge placed in front of a linear CCD array. The shadow edge caused by a light attached to the user was used to determine one degree of freedom. For sensors of 256 elements, the working volume was about 1 cubic meter with accuracy of 6 millimeters. This method could be used to track multiple lights by time multiplexing but had to trade off the number of lights against the sensitivity of the CCD array, the brightness of the lights, and the speed that the lights are blinked. Like the above systems, this method cannot directly determine orientation.

A new lateral-effect photodiode based system developed by the Microelectronics Systems Laboratory of UNC Computer Science allows a working volume of about 6 cubic meters with accuracy of about one part in 1000. It determines position in the same manner as the SELSPOT system but uses a new polarization method developed by the Microelectronic Systems Laboratory to directly sense orientation. The target is a light source placed under a cone of polarizing material made so that the angle of polarization of a ray of light can be used to determine the orientation of the assembly. A rotating disk of polarizing material placed in front of the fixed lateral-effect photodiode assembly modulates the incoming ray with a phase shift that is measured by additional circuitry. The major problem is that a very bright light surrounded by a cone of polarizing material must be attached to a target that is to be tracked. This causes problems of power delivery and heat removal and also of user distraction when the light enters his field of view. Another problem is that it is limited to tracking single objects.

2.2 New Tracking Method

The result of this research is a new tracking method that determines the position and orientation of a device that is held or worn by a user. The new method has been named SELF-TRACKER because the part of the system being tracked is also the part doing the tracking—it is “self” tracking. The most general characteristics and advantages of SELF-TRACKER are described first, followed by descriptions of some possible implementations.

SELF-TRACKER is optical. An optical system can be made less restrictive than mechanical systems, and more robust than acoustic or magnetic systems. Also, good optical detectors and lenses are more readily available and less expensive than ultrasonic microphones and high-precision magnetic devices.

SELF-TRACKER uses an outward view; instead of looking *in* toward the user from fixed places in the environment (Figure 2.1), the SELF-TRACKER cluster looks *out* toward the

environment from the user's position (Figure 2.2). Previous systems for real-time three-dimensional computer tracking place the sensors, which are large, at fixed locations in the environment and track light emitted or reflected by devices (LED's or mirrors) attached to the user. SELF-TRACKER places the sensors, which are small, in a rigid, baseball-sized cluster that is held or worn by the user. This cluster is somewhat like a fly's eye with a dozen or so facets each implemented with a small lens and a custom integrated circuit that combines the function of a fast camera with circuitry for measuring and reporting some image characteristics (e.g. image-to-image correspondence or positions of known beacons). The output of all the sensor chips is collected and analyzed by a separate general-purpose computer to determine the position and orientation of the cluster.

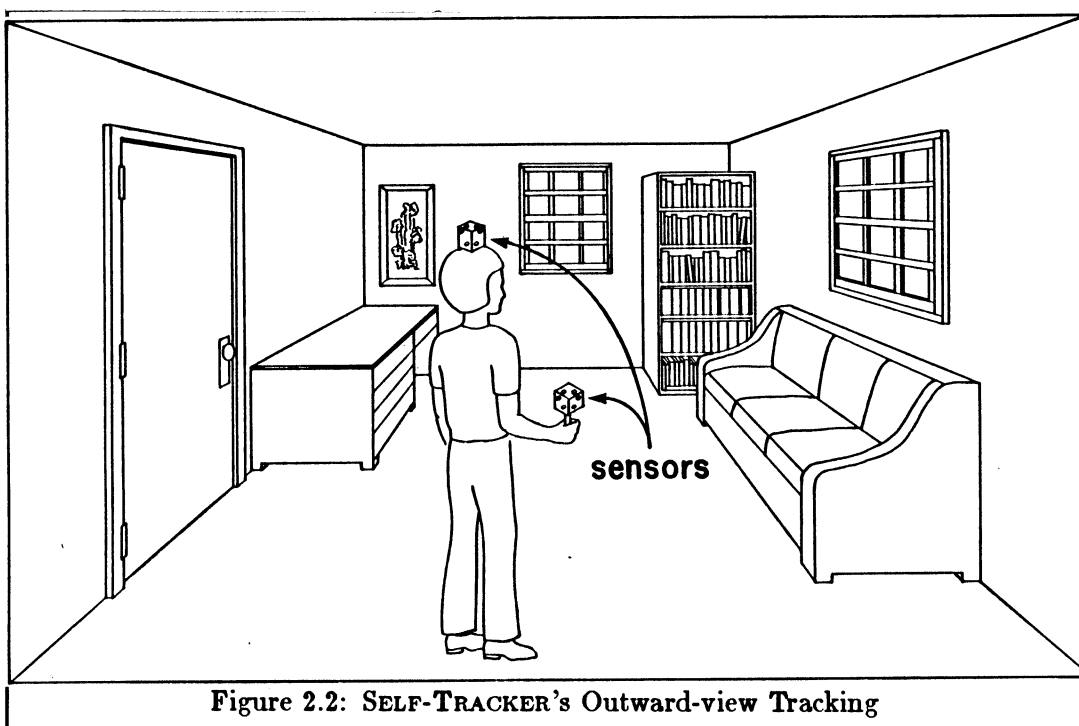
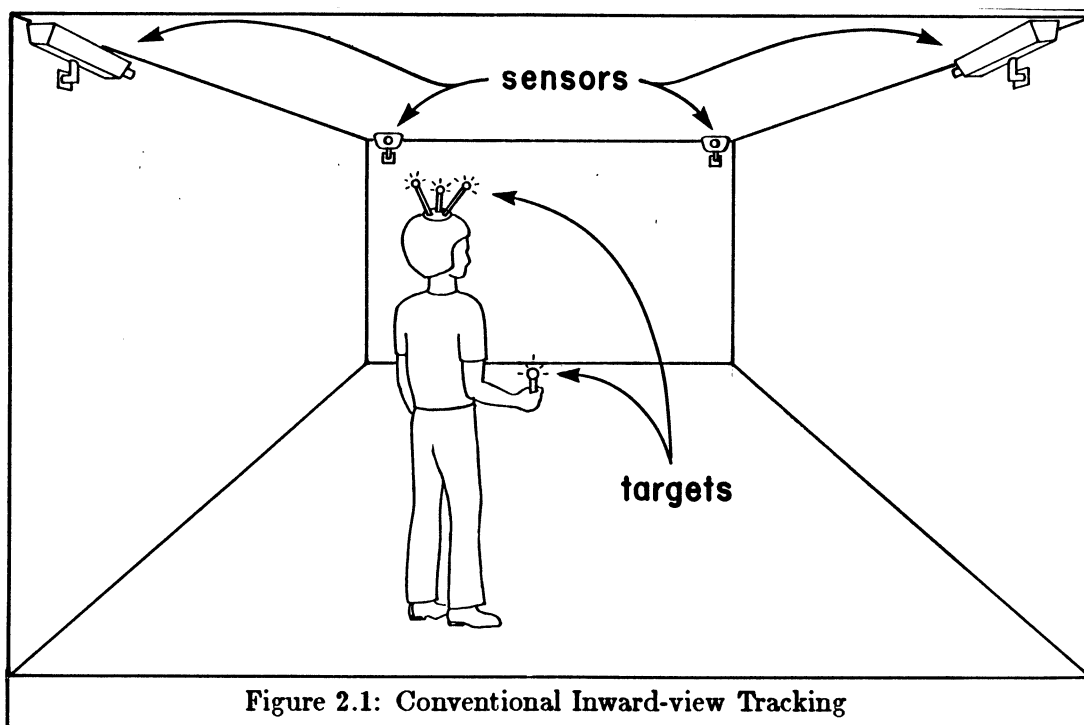
SELF-TRACKER is more portable than other tracking systems because the sensor assembly is small and because its rigidity allows calibration to be done once. Conventional tracking systems with the sensors fixed to the environment require mounting the sensors on walls or stands and they must be recalibrated whenever any sensor is moved. The SELF-TRACKER is calibrated once during assembly.

The SELF-TRACKER sensor cluster is passive; it neither generates radiation nor requires that radiation be directed toward the user. This eliminates the possibility of the user being distracted or injured by the radiation source.

SELF-TRACKER senses both position and orientation. Unlike optical systems with sensors fixed to the environment, SELF-TRACKER determines position and orientation from the data supplied by a single small cluster. The rigidity of the cluster is analogous to the fixed spatial relationship of the targets required by fixed-sensor systems.

2.3 Alternative Implementations

The remaining characteristics of SELF-TRACKER are determined by characteristics of the environment that the sensors observe. One implementation would have sensors that are specialized for detecting beacons that are at fixed locations. Another would have the sensors observe the natural room environment under normal lighting conditions. A third possible implementation would combine the first two in a system that observes the natural environment but also uses beacons to improve accuracy and stability. The next sections describe these three possible implementations.



2.3.1 Tracking with Beacons Only

An outward-view tracking system could be built that relied solely on beacons for its “navigation”. The system would combine high-resolution sensor arrays, lenses that provide a fairly wide field of view, and beacons at fixed locations in the environment. The beacons should be easily distinguished from the rest of the environment (e.g. blinking LED’s, or concentric circles of alternating reflecting and non-reflecting material). This implementation of the SELF-TRACKER would determine its position from the apparent position of the beacons, using a method similar to that used in photogrammetry.

Standard photogrammetric methods require sightings of at least four landmarks in a two-dimensional picture to determine the position and orientation of the camera. One-dimensional pictures would more likely be used in this implementation because one-dimensional sensor arrays allow much closer pixel spacing and thus better resolution than do two-dimensional arrays (see section 3.2.2); a one-dimensional array can be designed on the two-dimensional chip surface as long thin cells; the two-dimensional array elements must be square. If the photogrammetric method is extended to work with one-dimensional images from a cluster of sensors, seven beacons must be visible to determine the position and orientation of the cluster. (The requirement for seven beacons is a guess. I have not yet developed a tracking method for a beacons-only SELF-TRACKER. At least six will be required because there are six degrees of freedom in the clusters motion and each sensor provides at most 1 independent data point. Seven known points are required for calibration of a one-dimensional camera system (e.g. [Burton, 1973; Fuchs *et al.*, 1977]) and this is almost identical to determining the position of the SELF-TRACKER.) A filter that uses information about the past position of the cluster and restrictions on its possible motions (e.g. a Kalman filter) could be used to allow proper operation for short periods with fewer than seven beacons visible but for reliability and accuracy the system must be designed so that seven or more beacons are visible essentially all the time.

Having seven beacons visible at all times requires a compromise between the accuracy of the system, which is best with a small field of view, and the number of beacons, which is minimized by a large field of view. For best system accuracy a long focal length, yielding a small field of view, is preferred. But a small field of view requires that the beacons be closely spaced to assure that enough of them will be seen; many hundreds of beacons could be required. A large field of view reduces the number of beacons required, but it also reduces the accuracy and can complicate the design of the sensor chips if a single sensor array has to handle multiple beacons simultaneously.

To study this compromise between accuracy and the required number of beacons, I wrote a simple computer program that does a Monte Carlo simulation of a beacons-only SELF-TRACKER in a room with beacons on the walls and ceiling. Input to the program includes the size of the room, placement of beacons, number of sensor elements in the cluster, and the field of view of the sensor elements. Output from the program is a histogram indicating the number of beacons visible to each sensor element and to the entire cluster with the cluster at randomly generated positions and orientations. One typical simulation showed seven or more beacons visible at 73% of the 10000 random positions tried in the room. I have been unable to characterize the shape and position of the *dead-zones*, the percentage of the space in which too few beacons are visible, in the six-dimensional search space. I will do more research on beacon placement and field of view as a first step in continuing this design. It may be possible to characterize the dead-zones analytically allowing determination of best beacon placement and sensor field of view without trial and error.

Since tracking depends on sightings of these special beacons, the sensor chips and beacons must somehow work together to allow the beacons to be separated from other features of the background in the sensor's field of view. The simplest arrangement would have the beacons much brighter than anything else in the environment. The sensors could report a beacon sighting when a small group of photosensors switch significantly earlier than all others. This might not be reliable because room lights, reflections, or sunlight through windows might cause false sightings. Also, the beacons would have to be very bright to be significantly brighter than ambient in a normally lighted room. A better approach might have the beacons blinking at a high rate, and the sensors designed to observe several cycles before reporting a sighting. A third possible implementation would utilize ambient light and small "road signs" made of alternating strips of reflecting and non-reflecting materials. The sensors would detect the pattern of alternating light and dark regions and would report the estimated position of the center. This design would be complicated by the apparent variations in size of the pattern caused by changes in range and by the need to detect it in any orientation.

Another difficulty with beacons in this implementation is identifying the particular beacon that a sensor is reporting. One solution is to rely completely on information about the previous position and maximum expected motion to determine which beacon a particular sensor sees; if the beacons are sufficiently far apart, this method could be reliable. Another solution would encode the beacon's identity in its blink rate, its duty cycle, or in the relative sizes of bright and dark regions (as in the universal product code).

This encoding does not have to be unique, since history can be used if adjacent beacons have different codes. Another, much less desirable, solution would have all the beacons controlled by the general-purpose computer. It could positively identify them by turning them on and off but it would be much more complex because of the large amount of communication required to control the beacons.

2.3.2 Tracking in a Natural Environment

Another SELF-TRACKER implementation would solve the problems of the beacons-only system by eliminating the beacons completely. The sensor chips in this implementation each examine a different small view of the room and report the apparent image changes caused by the motion of the cluster. The sensor chips also measure the distance to the scene. A separate general-purpose computer uses the image change and distance information to extract the three-dimensional motion of the cluster.

It is important to note that this implementation measures *motion*, not *position*. To determine the position and orientation of the cluster, the analysis algorithms must integrate the motions from an initial starting position. Error will, of course, accumulate with time, resulting in drift in the measured position. Small errors are tolerable in a head-mounted display system if the displayed objects do not have to register with real objects in the room but a method of compensating for the drift must be found before this implementation could be practically used. The error behavior of this SELF-TRACKER implementation is similar to that of inertial guidance systems but SELF-TRACKER measures velocity rather than acceleration. Error accumulation should be less severe with SELF-TRACKER because we are solving a first-order differential equation rather than a second-order one.

The major problem in the design of a natural-environment SELF-TRACKER is measuring the image change seen by each of the sensors in the cluster. The chips must analyze successive images of a small view of the room to determine how the image has changed; this is a difficult problem in general, as evidenced by the substantial literature on the correspondence problem (e.g.[Ullman, 1979; Ballard and Brown, 1982]). The solution proposed in chapter 3 is to operate at a high frame rate, 1000 to 4000 frames per second, rather than the 30 frames per second commonly used in video systems. Because this frame rate is high in comparison to the speed of human motions, there is little change in successive images. With small image-to-image changes, a simple measurement technique may be able to measure the image shift accurately. Since the simple technique can be implemented with simple and small circuitry, the entire sensor including the high-resolution photosensor array and processor for registering successive images can fit on a single chip. This allows

high-speed operation because none of the critical signals must be driven off the chip. This circular relationship with speed producing simplicity producing speed is discussed in section 3.2.1.

This short frame time is critically dependent on the sensitivity of the photosensors; the system cannot process the image until the photosensors make it available. I have designed, fabricated, and tested photosensors that operate at 500 to 5000 frames per second in ordinary room light with a small lens. These photosensors are described in chapter 5.

Another important factor in the design of a natural-environment sensor is the resolution of the sensor. The ability of the sensors to detect small translations and rotations depends on the spacing of the photodiodes in the sensor and on the focal length of the lens. This focal length cannot be made arbitrarily long to improve system accuracy, because the scene must contain useful features and these features must be visible in successive images to allow measurement of shift. If the focal length is made too long, thus making the field of view very small, the sensor may not see any features (for example, only a small patch of smooth wall surface may be visible). Also, the frame time would have to be very short to maintain continuity between images at high rates of rotation. Experiments with digitized video images of our laboratory and analysis of system accuracy (section 3.2) indicate that a 10 degree field of view can provide images with adequate features for registration and can provide reasonable system accuracy.

Earlier in this section I mentioned that the sensors must measure distance to the scene as well as motion. The distance is required by the three-dimensional motion extraction algorithm (section 3.3) to restore the scale for translations that is lost because of perspective distortion (section 3.1). SELF-TRACKER measures distance using stereo pairs of the motion sensor chips. The same circuitry that is used for measuring image shifts caused by motion, measures the shift caused by the spatial separation of the chips in the stereo pair. This requires only a small circuit for asynchronous communication between the chips (section 3.2.4) and about a dozen more instructions in the chip's program.

2.3.3 A Combined System

A third implementation of SELF-TRACKER would combine beacon tracking with motion tracking to provide a system with the best features of both. This implementation would not require a large number of beacons because the motion tracking system would provide sensitivity to small motions and would eliminate problems with dead-zones. Also, this

implementation would not experience long-term error accumulation, because the beacon tracking subsystem would provide absolute fixes.

The design parameters of a combined system are the same as those for the two independent systems, but now the compromises that must be made are different. The beacon system can use fewer beacons because dead-zones are now not as important. Also, one could design an extraction algorithm that did not require as many beacon sightings by combining the absolute and relative measurements in a single mathematical model. The beacon sensors could use lenses of shorter focal length, thus providing a larger field of view, because high resolution is not important in the beacon subsystem since the motion system can measure the small motions. The motion system is improved as well. It could use lenses of longer focal length to get higher resolution without the problem of losing context during fast rotations.

The cluster could be implemented with different chips for the beacon sensing and the motion sensing, allowing the subsystems to be developed separately with little design interaction. Once the systems were developed, they could be integrated into a cube shaped cluster with three lenses on each face, two for the motion sensor pair and one for the beacon sensor.

2.4 Choice of a Method for Further Study

I chose the natural-environment SELF-TRACKER for further study because it offered the most interesting design problems and because it is required for the combined tracking system. The photosensor design developed for the natural-environment tracker will probably be used in the beacon tracker, and the general-purpose control computer and interface for the natural-environment tracker can be used for both the beacon tracker and the combined system.

Chapter 3

A Natural-Environment Tracker

This chapter is an analysis of the design of a natural-environment SELF-TRACKER. It includes the design equations for the optical systems of the cluster, and some suggested cluster designs. Also, the simplifying assumptions made in the design of the sensor chips are described and justified. The chapter concludes with results from a simulation I did to study error accumulation in the proposed design.

3.1 Design of the Cluster

3.1.1 Goals for the Cluster Design

The first goal for SELF-TRACKER is sufficient accuracy for use with a head-mounted display. This accuracy depends on the types of images presented in the display. If the images are justified to the environment (for example a computer generated cup on a real table) the system must be very accurate to prevent apparent motion of the cup relative to the table. If, on the other hand, the images are not justified to the environment (for example a model of a molecule floating in air) it would seem that a larger amount of error could be tolerated. I have characterized tracking errors by the errors they cause in the displayed image of a point 1 meter in front of a user with a helmet display that produces a 90 degree field of view of a 512×512 pixel image.

The second goal is operation in a room-sized environment. The head-mounted display system we want to build at UNC will allow the user to move around in a large space so that he can interact with large images in a natural manner. A space 4 meters on a side and 2.5 meters tall was chosen as typical of office-sized environments.

The third goal is small size and light weight. Weight is particularly important because the cluster will be mounted on a helmet or on the end of a hand-held sceptre. Size is important for practical use (a basketball on the end of a sceptre would not be very useful) and will allow rugged construction. An orange is about the size I would prefer for the cluster; a grapefruit is about as large as I could tolerate.

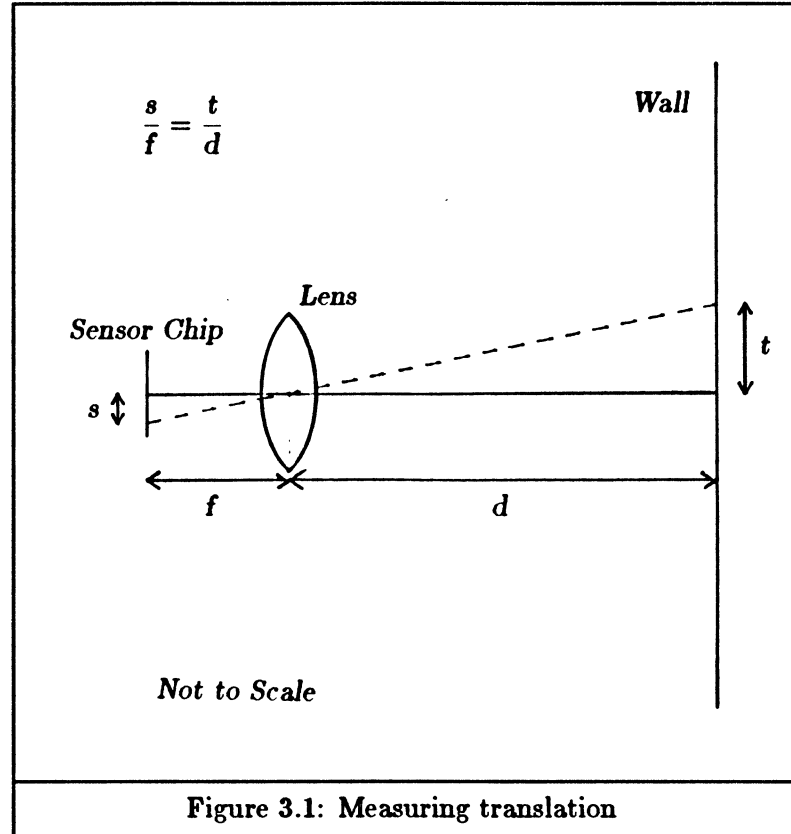


Figure 3.1: Measuring translation

3.1.2 Design Equations for the Cluster

Throughout this section, the distance to a scene is assumed to be much greater than the focal length of optical system being discussed. This allows use of the simple pin-hole camera model for the optical systems. This assumption is valid for the proposed designs with short focal lengths, less than 50 millimeters, operating at distances of 0.5 to 4 meters.

Measuring Translation. Figure 3.1 is an example of translation measurement with the SELF-TRACKER. To simplify the figure, the room is shown translating rather than the sensor; these are, of course, equivalent. The shift, s , measured by a sensor with a lens of focal length f for a translation, t , at distance d is given by

$$s = \frac{ft}{d}. \quad (1)$$

It is important to notice that the measured shift depends on the distance to the scene. A larger translation at a proportionally larger distance would produce the same shift at the sensor. Later sections will show that this dependence on range has a profound affect on the SELF-TRACKER's design and its accuracy.

The translation can be estimated from the measured image shift (\hat{s}) and distance (\hat{d}) by substituting for s and d and solving for \hat{t} .

$$\hat{t} = \frac{\hat{s}\hat{d}}{f}. \quad (2)$$

Now \hat{s} and \hat{d} differ from their true values, s and d , because of the limited resolution of the sensor. The bound on the relative error in \hat{t} is computed from the relative error bounds in \hat{s} and \hat{d} using the methods in Pizer and Wallace, 1983.

$$r_{\hat{t}} = r_{\hat{s}} + r_{\hat{d}}. \quad (3)$$

The relative error bound for distance, $r_{\hat{d}}$, will be determined in the next section. The maximum error in shift \hat{s} , assuming that the shift can be measured within one pixel, is one half the pixel spacing, p , so the relative error bound is

$$r_{\hat{s}} = \left| \frac{p}{2s} \right|. \quad (4)$$

But s , the true value of the image shift, is given by (1) above so

$$r_{\hat{s}} = \left| \frac{pd}{2tf} \right|. \quad (5)$$

This is one important result of the dependence on distance to the scene—the error in measured translation grows directly with the distance to the scene. Obviously, the SELF-TRACKER will not be practical in environments that are too large. This relative error can be quite large (8% for a 5 mm translation at 2 meters with a 45 mm lens and 13.5 μm pixel spacing) but this is for a single sensor. Combining measurements from several sensors should reduce the error (section 3.3).

The obvious method to reduce the translation error is to increase the focal length, but the field of view of a sensor will grow too small to include any significant detail if the focal length is made too long.

A better method for reducing error exploits the reduction in relative error as the measured motion gets larger. The motion between frames can be increased by decreasing the effective frame rate. Rather than always comparing successive frames, compare every other frame or every fifth frame based on the amount of shift measured. If the measured shift between frame i and frame $i+1$ is small (e.g. 1 pixel or less) and the disparity is low, keep frame i rather than $i+1$ for comparison to frame $i+2$. Whenever the shift or minimum disparity exceeds some threshold, abandon the old frame and latch the new one. This

approach allows the effective frame rate to adapt to the speed of the users motion without sacrificing the ability to respond to sudden motions. For example, a SELF-TRACKER sensor that always compares successive images and that operates at 1000 frames per second, with a 45 mm lens and $13.5 \mu\text{m}$ pixel spacing, will never register any motion if the user is moving at 30 cm per second or less while 2 meters from a wall. The sensor can not register motion because the image shift in 1 millisecond is less than one-half pixel. If the method described above is used, the system could allow the image shift to accumulate to the point that it would be measurable. If images that are 10 frames apart are compared, the minimum measurable rate of motion drops to 3 cm per second; 100-frame separation allows measurements down to 3 mm per second.

Measuring Range. To determine translations from image shifts the distance to the scene must be known. A method for measuring this distance can be derived from equation (1) by solving for d .

$$d = \frac{ft}{s}. \quad (6)$$

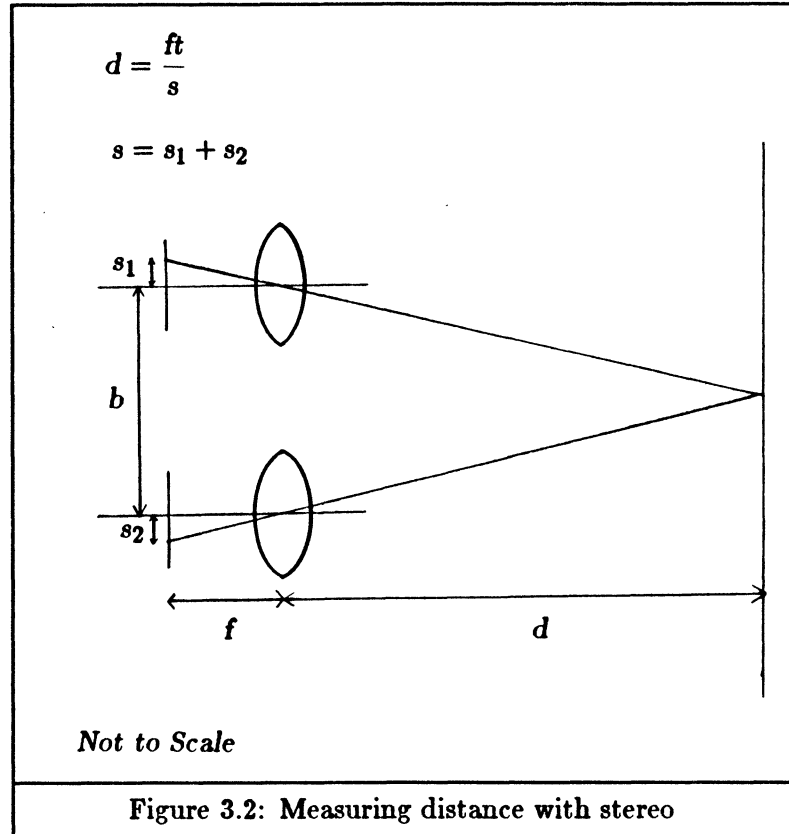
The focal length is known and the image shift is measured but somehow the translation must be known if distance is to be determined. The translation will be known if two sensors with a fixed *base-line* separation are used. Figure 3.2 shows distance measurement using stereo separation. The same method for measuring image shifts can be used to measure distance as well as motion. The only change to the sensor chips is the ability to transfer images from one sensor chip to another.

The error in measured distance is characterized using the bounds analysis again.

$$r_d = \left| \frac{pd}{2bf} \right|. \quad (7)$$

Notice that the *relative* error grows with the distance; the *absolute* error will grow as the square of the distance. Happily, the distances to be measured in an office-sized room are around 2 meters, so the contribution of this term is small for practical values of base-line separation and focal length. With the parameters used in the translation example (45 mm lens, 2 meters to the scene, $13.5 \mu\text{m}$ pixel spacing) and a base-line separation of 50 millimeters, the relative error bound on the measured range is 0.6% (1.2 cm maximum error).

Improving the accuracy of range measurement by increasing the focal length or base-line separation is limited by the size of the cluster and by the minimum distance that the stereo pair can measure. Increasing either the focal length, or the base-line

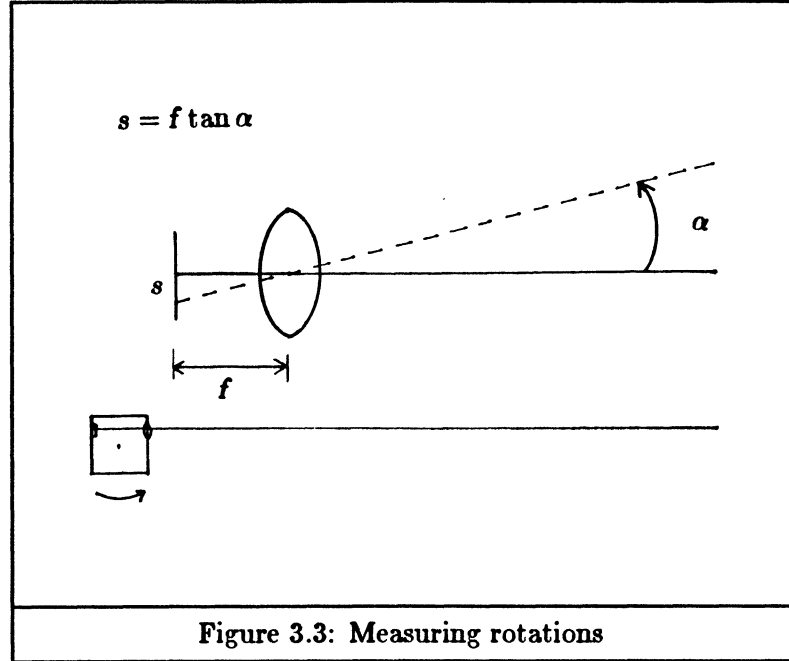


increases the shift between the image pairs, thus decreasing the relative error, but there is some maximum shift that allows a reliable match. If the images overlap less than about 30 percent, the number of image features available for shift measurement may be too small for good reliability. The minimum measurable range, d_{\min} , is related to the base-line, b , the focal length, f , the pixel spacing, p , the width of the sensor array in pixels, N , and the minimum acceptable overlap, O_{\min} by

$$d_{\min} = \frac{bf}{Np(1 - O_{\min})}. \quad (8)$$

The minimum measurable range for a minimum overlap of 50 percent with the same parameters again and with 400 pixels in the array is 0.8 meter. Sensors that are closer than this minimum range to the scene may produce inaccurate distance measurements and will have to be ignored by the control computer.

Measuring Rotations. Rotation measurement is much less error-prone than either measurement of translation or range, because rotations are independent of range. As



shown in Figure 3.3, the shift reported by a sensor for a given rotation is determined solely by the amount of the rotation. The equation for rotations is

$$\alpha = \tan^{-1} \frac{s}{f}. \quad (9)$$

The relative error bound on the measured rotation, $\hat{\alpha}$, for small α , is

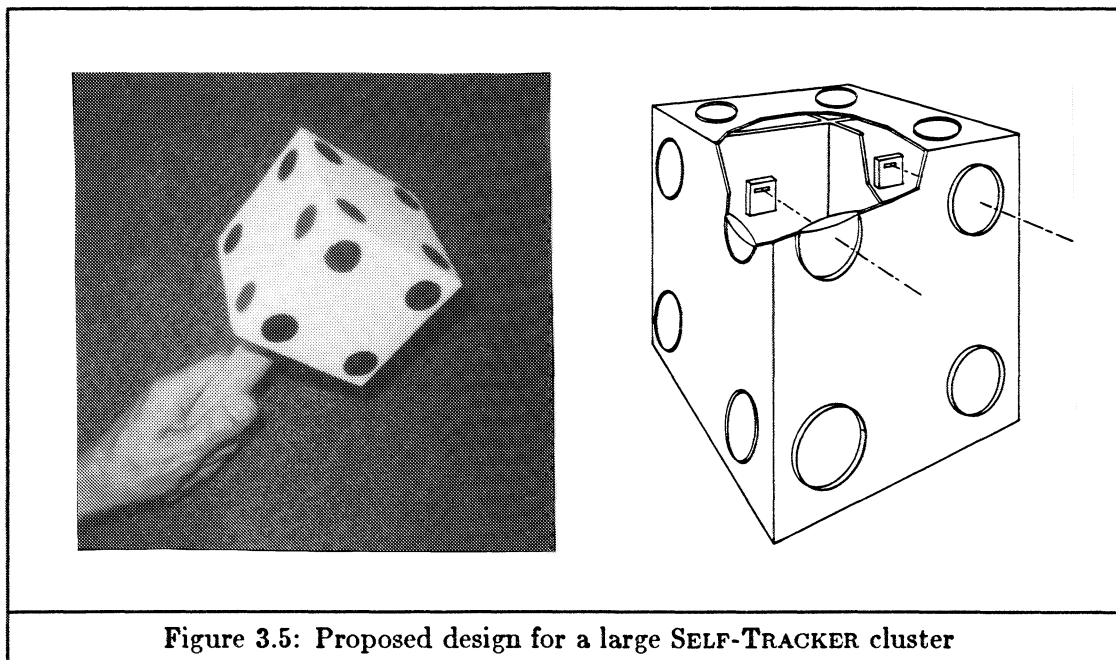
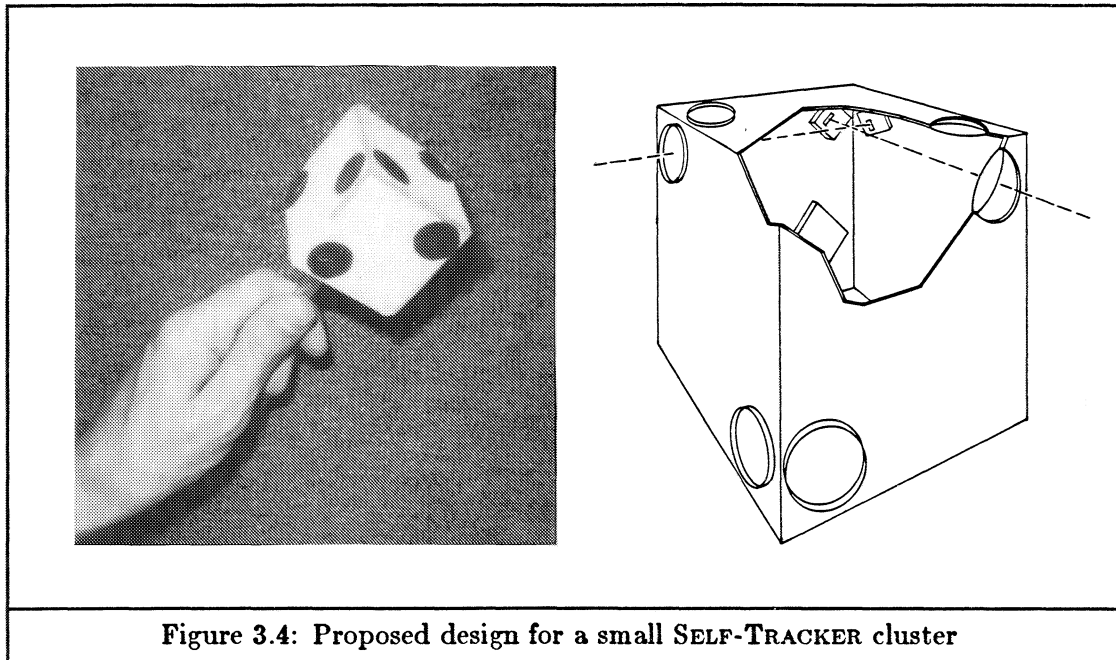
$$r_{\hat{\alpha}} \approx \left| \frac{p}{2f\alpha} \right|. \quad (10)$$

The relative error for a rotation of 1 degree with the parameters used in previous examples (45 mm lens, 13.5 μm pixel spacing) is 0.8%. This error can be reduced by using the adaptive comparison method described earlier for reducing the error in translations. For example, a rotation of 5 degrees would have a relative error of 0.16%.

3.1.3 Proposed Cluster Designs

A small SELF-TRACKER design would mount lenses and sensor chips on opposite sides of the cluster enclosure, Figure 3.4, so that cluster size is determined by the focal-length. This design is highly constrained since the focal length and base-line must be compatible and obscuration must be avoided.

A larger SELF-TRACKER design with twice as many sensors for more redundancy would make the cluster diameter a little more than twice the focal length by mounting the sensor chips in the center as shown in Figure 3.5. This design is less constrained since the base-line is largely independent of the focal length.



3.2 Design of the Sensor Chip

3.2.1 Frame Rate and Simplicity

In the previous discussion, the sensor chips were assumed to measure changes in successive images without specifying how this is done. Reference to the extensive literature for the case of successive images from a single television camera shows that this is hard (e.g. [Ullman, 1979; Ballard and Brown, 1982]). Implementation on a single chip, as required for SELF-TRACKER seems unreasonable unless the problem can be simplified.

For example, suppose a system acquires an image and measures the change since the last image in 10 milliseconds for image changes corresponding to 1 degree of rotation. The experiments that I will describe later in this section indicate that as much as 10 degrees of rotation can occur in 10 milliseconds at peak rates of head rotation and even at normal rates, 2 degrees of rotation could occur. But the 10 millisecond method is good only for 1 degree rotations, 10 degrees might require 100 milliseconds or more. But now, all is lost. In 100 milliseconds the user could be virtually anywhere. He could have rotated his head as much as 100 degrees; so much that the images to be compared are of completely different scenes.

On the other hand, a system that requires 1 millisecond for image changes corresponding to 1 degree of rotation can just keep up, and a system that requires less than 1 millisecond can do better. If measuring 1 degree changes requires only 0.5 millisecond then only 0.5 degree changes must be measured but this might require only 0.25 millisecond so only 0.25 degree changes are measured. At some point this decreasing time spiral will stop because only the time to process the image is decreasing; the time to acquire the image is fixed by the available light and the sensitivity of the imager.

The first step toward image-to-image times of 1 millisecond or less is photosensors that are sufficiently sensitive for operation at this rate in ordinary room light. I have designed and tested photosensors that allow image-to-image times of 200 microseconds or less (see chapter 5).

The second step is an image-change measurement method that can keep up. Sufficiently fast operation has been achieved by making two simplifications. First, the high frame rate assures that the human user cannot have moved very far between images, thus the images are very similar and a simple image comparison method can work reliably (section 3.2.3). The second simplification follows from using multiple identical sensors looking in different directions, rather than a single camera. The use of multiple redundant

sensors allows each sensor to ignore image rotations and changes of scale because motions that generate a rotation or change of scale for a sensor looking in one direction will produce a simple translation in one of the others that is looking in different direction.

How fast must the sensors operate to assure small image-to-image changes? To find out, I measured the rate of head motion by examining successive frames from a standard television camera. I used our Ikonas RDS3000 and a simple microcode program written in Gia2 [Bishop, 1982] to digitize and store 50-line-by-100-pixel frames in the large frame buffer memory in real-time. The program can record 600 fields of video at 60 fields per second allowing, 10 seconds of recorded action. My measurements gave 70 degrees per second as the natural rate for most head rotations and 750 degrees per second as the peak rate.

To allow a margin of safety, I have chosen 200 degrees per second as the rate for normal head rotations and 1000 degrees per second as the peak rate. These rates are supported by industrial time and motion literature [Quick *et al.*, 1962], which places the rate at 188 degrees per second (45 degrees in 240 ms). With the cluster design used previously in this chapter (45 mm lens, 13.5 μm pixel spacing, 400 pixels), the required frame rate can be estimated from the maximum allowable image-to-image change. Assuming a maximum allowable shift of 20 pixels, the required rates are 580 frames per second at 200 degrees per second and 2900 frames per second at 1000 degrees per second. The prototype SELF-TRACKER 1.0 chip described in chapter 4 operates at 1000 frames per second. Increasing the size of the photodiodes and fabrication with 3 μm features should allow operation at 3000 to 4000 frames per second.

Previous Work by Others. The basic idea of using fast, smart sensors is not original with this research. Richard Lyon's Optical Mouse [Lyon, 1981](see chapter 1) was the first of this new type of sensor. He combined a 4×4 pixel square sensor array with processing circuitry to follow the motion of a special dot pattern. Howard Landman used a 256 pixel linear sensor array to track the motion of a guitar string for an optical pickup [Landman, 1983]. John Tanner used a linear array of 16 pixels and an analog plus or minus 1 pixel correlation circuit to make an optical mouse that follows any pattern (e.g. wood grain on the desk top) [Tanner and Mead, 1984]. Herbst, Grassl, and Pfeleiderer [Herbst *et al.*, 1982], used two 24 pixel arrays to measure distance via stereo for an experimental autofocus control for a 35 mm camera.

The SELF-TRACKER is another step in the development of smart optical sensors on silicon. SELF-TRACKER uses a high-resolution linear sensor and high-speed processing to

measure both motion and distance of natural scenes with a single custom chip design. My design is substantially more complex than any of these others because higher speed operation is required with less light and with more complex images.

3.2.2 Imaging System

One-Dimensional Images. A two-dimensional imaging system was my first choice for the natural-environment SELF-TRACKER because I could measure shifts along two axes with a single sensor and because I am accustomed to working with two-dimensional images in graphics and image processing. This choice was reconsidered when I realized the importance of pixel spacing in the design equations given earlier in this chapter. In every case, reducing the pixel spacing improves the error performance of the system. Unfortunately it is not possible to achieve close spacing in a two-dimensional sensor array on the two-dimensional surface of a chip, because a minimum photodiode area is required to achieve the desired sensitivity and the amplifier must be close to the photodiode. I estimate that the minimum possible pixel spacing for a two-dimensional array with $3\text{ }\mu\text{m}$ minimum circuit features is $100\text{ }\mu\text{m}$ (see Figure 3.6). This pixel spacing would require impractically long focal lengths to achieve adequate system accuracy. The cluster would be too large and the field of view too small to assure the presence of adequate image features for shift measurement.

The pixel spacing can be reduced by using a one-dimensional imaging system composed of a linear array of photosensors. A one-dimensional array can be designed with close pixel spacing by making the photodiodes long and thin and by placing the amplifier and support circuitry for alternate pixels on opposite sides of the array (see Figure 3.6). I have designed an array that would have $13.5\text{ }\mu\text{m}$ pixel spacing with $3\text{ }\mu\text{m}$ features. The array has been implemented and tested with $4\text{ }\mu\text{m}$ features and $18\text{ }\mu\text{m}$ pixel spacing. This array operates at from 500 to 5000 frames per second and includes circuitry for forming an image, and for shifting the image out to the processor.

It was not obvious that one-dimensional pictures of natural scenes would consistently provide enough image features to measure image shifts. It seemed that the loss of one-dimension might obscure all useful detail. To determine the usefulness of one-dimensional images of natural scenes, I simulated the operation of a photosensor array using a video digitizer and a standard television camera (IkonaS RDS 3000, RCA TC2000 camera, 75 mm zoom lens). I digitized scenes of our graphics laboratory with the lens set for a 10 degree field of view. I then generated one-dimensional images from the two-dimensional images by extracting a 256 by 50 pixel region and summing the intensities in the 50 pixels columns to

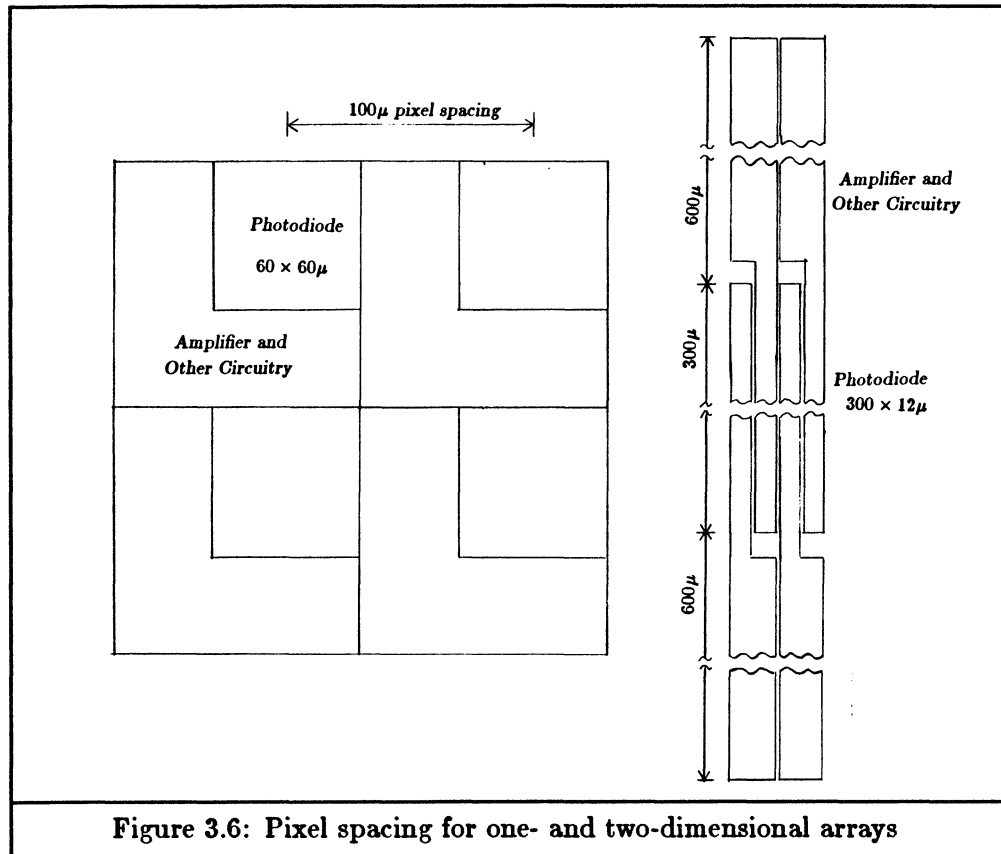
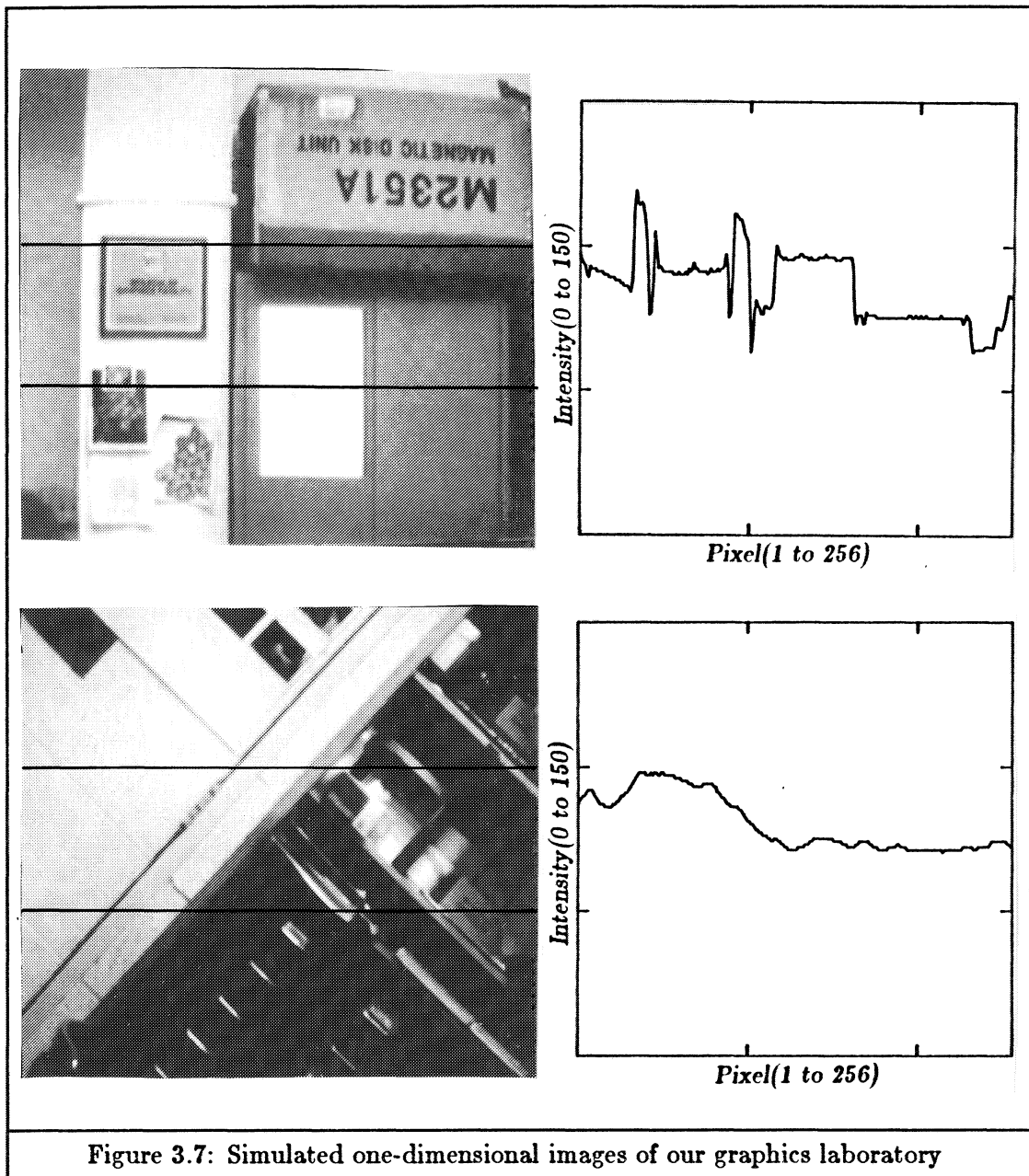


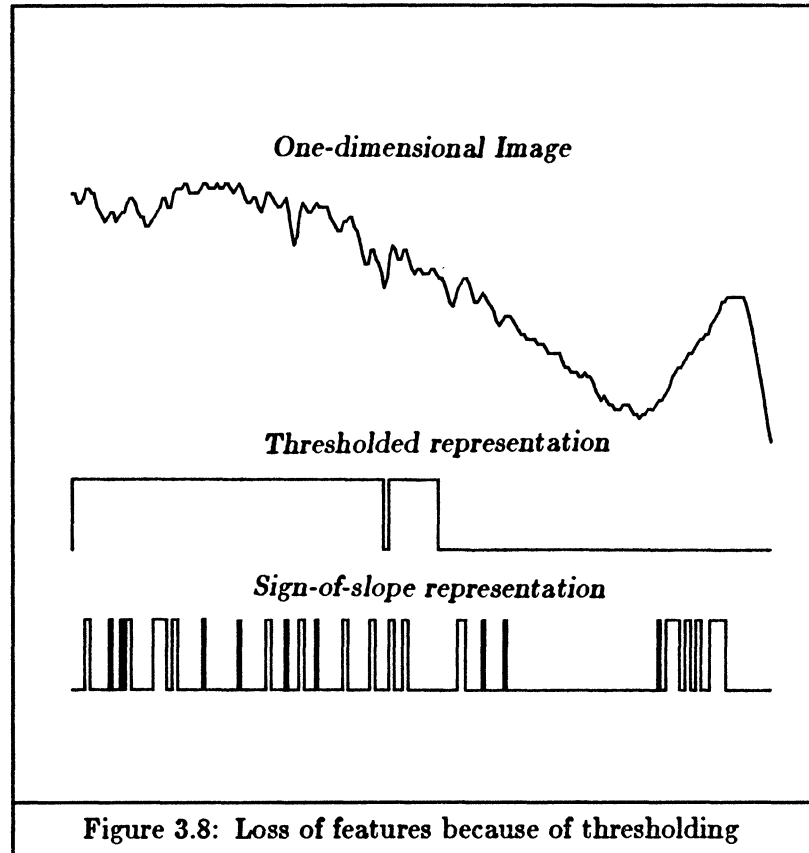
Figure 3.6: Pixel spacing for one- and two-dimensional arrays

derive a 256 pixel image. Figure 3.7 shows some scenes of our laboratory and the resulting one-dimensional intensities. The first image is typical of those a sensor chip would see, the second was carefully chosen to show one of the worst cases encountered (all features at 45 degrees to the sensor). Of course, much information is lost in the conversion and sometimes the one-dimensional image is not useful, but my experience with these simulations gives hope that this is rare. Also, the SELF-TRACKER cluster will contain many one-dimensional sensors at different orientations, so even if one or two of them cannot see significant image features, the others should supply the needed shift measurements.

Bi-level Images. The array of photosensors does not produce an image in the usual sense of an array on intensity values. The individual outputs merely switch from off to on after a delay determined by the intensity of the incident light. The relative switching times must somehow be used to form a useful image. One approach would repeatedly sample the outputs at intervals much shorter than the expected delays to form a multi-level image as normally used in image processing. I chose instead to use binary images to simplify the image comparison process.



A simple method for forming a binary image waits for some fixed number of pixels to switch and then latches the image. This thresholding of the image can produce very bad results for a large class of images. For example, the image in Figure 3.8 has several features that might be useful for shift measurement but it also has an intensity gradient. All of the useful features are lost in the thresholded binary image.



Some method that responds to local intensity changes is needed so that wide intensity variations over the image will not “swamp out” the important information. An attractive method would result in an image that has ones wherever the image intensity (thus the switching time) is “significantly different” at adjacent pixels and zeros elsewhere. I have been unable to devise simple circuitry that implements this because “significantly different” is relative to the absolute brightness.

Circuitry that produces an approximation to the above compares the relative switching time of pairs of pixels and produces a 1 if the left pixel is first and a 0 if the right pixel is first. This produces the sign of the slope of the intensity along the array. Significant features, such as a rapid increase in intensity in a region of gradually increasing intensity, can be lost, but my experiments with digitized images suggest that this method works well. A problem with this method that unfortunately was not discovered until after fabrication is that the small differences in sensitivity between adjacent sensors are also reported, resulting in patterned noise in the image. The differencing process needs to be stopped before small intensity variations become important (see section 4.2.1).

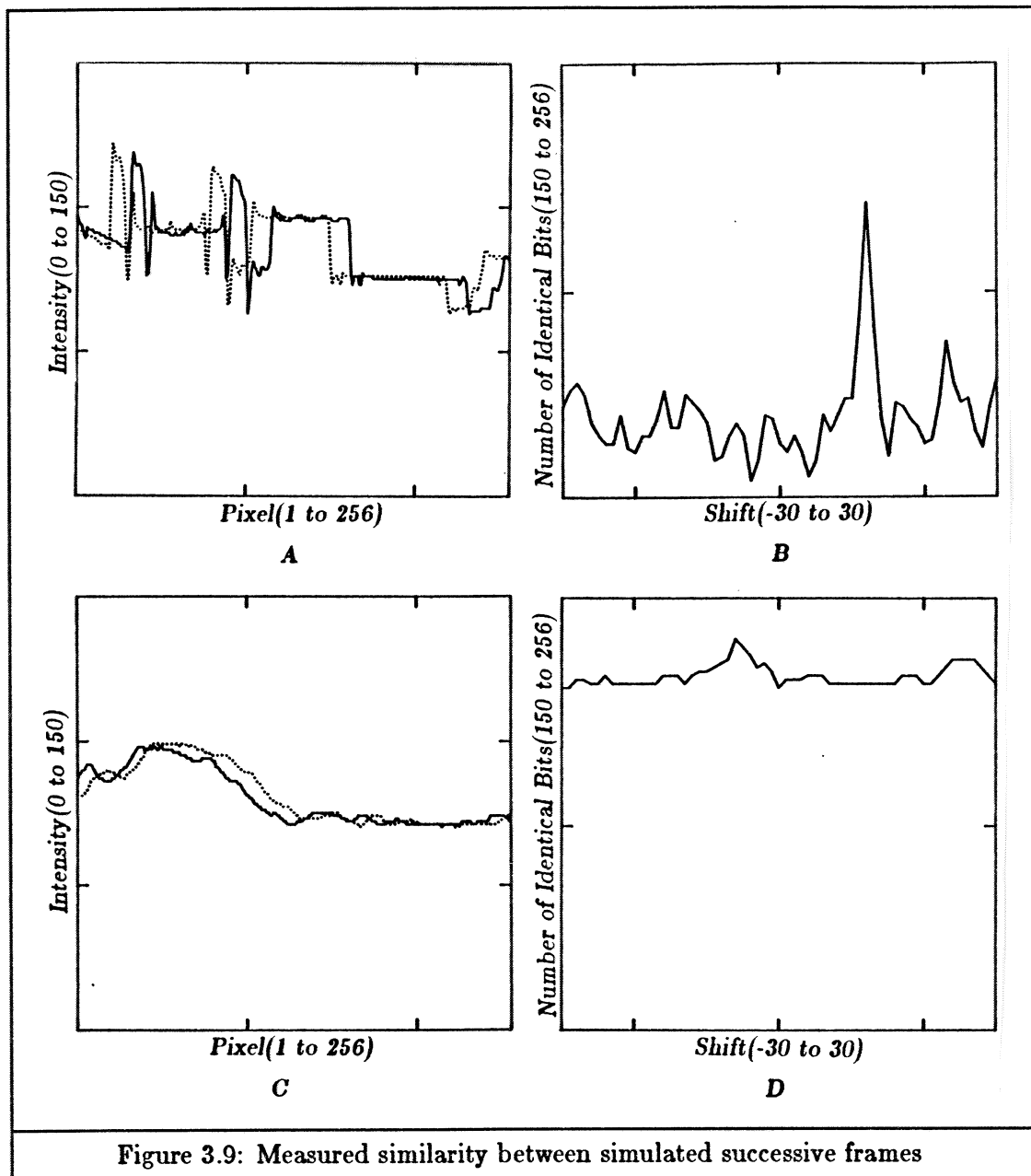
This sign-of-slope representation is well suited to the registration operation that follows because changes in intensity will often correspond to important image features such as the edge of a doorway or a seam between ceiling tiles. Multi-bit grey-scale images were not used because their processing would require more complex circuitry and because registering grey-scale images with simple methods like correlation often produces a broad maximum. Venot [Venot, 1983] has described a method for registering medical images that is similar to mine. He registers two grey-scale images by finding the relative shift that minimizes the number of sign changes in the difference of the two images. The number of sign changes has a sharp minimum at the correct image registration. Likewise, the number of identical bits in two sign-of-slope representations has a sharp maximum at correct registration.

3.2.3 Registration System

The image registration process is greatly simplified by the assumptions made in section 3.2.1; ignoring rotations and changes of scale eliminates much complexity that is related to extracting image features and matching them independently. The processor need only find the relative shift that maximizes some global measure of image similarity to determine the image shift. As mentioned earlier, sometimes these assumptions will not be valid and the measured shift will be incorrect, but motions that violate these assumptions for a sensor looking in one direction will be valid for another. To differentiate between good and bad matches, the processor will report the degree of similarity (the number of identical pixels) between the images as a measure of confidence of the match. The motion extraction algorithm running on the control computer will use the confidence measure as one factor in deciding which sensors to “believe” and which to ignore.

The search for maximum similarity should be done by exhaustive search over the range of possible shifts (about 10 pixels for motion measurements) rather than by hill climbing. Hill climbing is inappropriate for this application because there may be several local maxima in the search space.

Figure 3.9 shows the measured similarity (the number of identical bits in the bi-level sign-of-slope representation) of two pairs of one-dimensional images of our laboratory. The solid curve in 3.9A and C is the same as shown in Figure 3.7. The dotted curve was made by moving the camera to camera to simulate the image shift at a high rate of rotation with a one millisecond frame rate. The curves in 3.9B and D show the similarity at different relative shifts of the two images in A and C respectively. Before registration, the images were converted to the binary, sign of slope of magnitude, representation described earlier.



The large peak at maximum similarity is typical of the many scenes I have tried. Even images that were carefully chosen to contain few features show a peak at the correct shift.

Effect of Blur Caused by Motion. The images processed by the SELF-TRACKER will be blurred by the user's motion because the photosensor integrates light for a finite time interval (about 1 millisecond). The blur may limit the accuracy of the registration process, since nearly all of the shift occurred during the time the photosensor was integrating. I

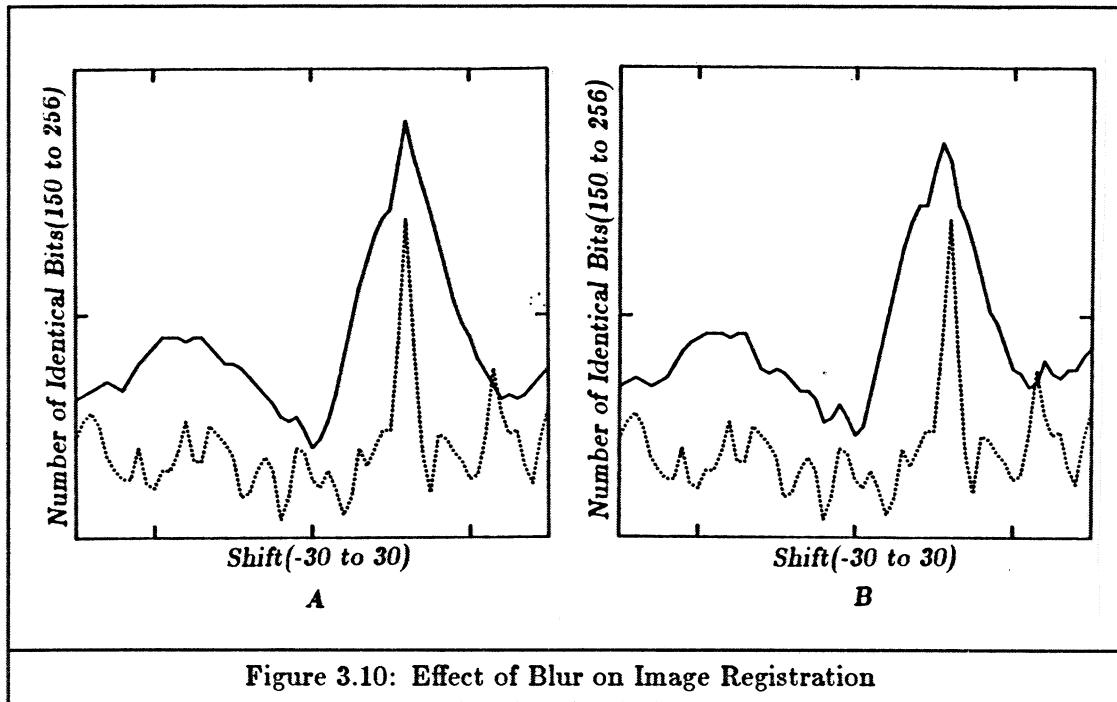


Figure 3.10: Effect of Blur on Image Registration

blurred the images in Figure 3.9A to see how well registration after conversion to a bi-level image would work. Figure 3.10 shows the effect of image blur on the image registration process. The similarity curve in 3.10A was computed by blurring the two images in 3.9A by 10 pixels each. The solid curve in 3.10A and B is the same as in 3.9B. The dotted curve in 3.10A is the result of blurring both the solid and dotted image in 3.9 by 10 pixels, simulating normal motion with no acceleration. The dotted curve in 3.10B results from blurring the solid curve of 3.9A by 10 pixels and the dotted curve by 8 pixels, simulating the effect of acceleration. Image registration works well (in fact better than I expected) in both cases, although the simulated acceleration in 3.10B results in some error. I believe the excellent performance results from “edge enhancement” by the sign-of-slope bi-level image representation (section 3.2.2).

3.2.4 Communication

The sensor chips must communicate with the control computer to report shift and confidence measurements and with their partner in the stereo pair to allow images to be shared. The communication must be asynchronous because each chip has its own internal clock. The data rate to the host will be about 2 kilobytes per second (8 bits of shift and 8 bits of confidence, 1000 times per second). The data rate to the other chip in the stereo pair will be about 50 kilobytes per second (400 bits of image data, 1000 times per second).

These rates should be simple to realize with standard serial (three wire) asynchronous links.

3.3 Design of the Motion Extraction Algorithm

The motion and range data provided by the individual sensors in the cluster are combined by a separate general purpose computer to determine the three-dimensional motion of the cluster. The extraction algorithm and supporting hardware must be designed to operate fast enough to keep up with the data flowing from the multiple sensor chips. With 20 sensor chips operating at 1000 frames per second the control computer will have to collect data at 40 kilobytes per second and solve for the motion of the cluster 1000 times per second. The linear approximate solution described in section 3.3.2 requires solution of a linear system of 20 equations in 6 unknowns which requires approximately 800 multiplies and adds, using the standard method of forming the normal equations by multiplying by the transpose of the coefficient matrix and solving the normal equations using Gaussian elimination. Floating point operations at 800,000 per second are beyond the capabilities of current supermicros and many inexpensive attached array processors. Array processors based on the new 5 MFlop floating-point chips from Weitek should be inexpensive and should easily solve the equations in under 1 millisecond.

3.3.1 Problem Formulation

Extracting the three-dimensional motion of the cluster from the shifts reported by the sensors can be very complicated in a completely unrestricted environment. For example, other people moving in the room might cause a sensor to report a false motion. Even in the absence of other people, a sensor might see the users hand or body and produce a false report. Also, sensors may occasionally produce misleading reports because of "pathological" scenes. I believe that these problems can be solved using a Kalman filter [Gelb, 1974; Liebelt, 1967], which would include a model of reasonable motions for the cluster so that invalid reports could be ignored, or RANSAC [Fischler and Bolles, 1981], a method for eliminating erroneous inputs by generating a model based on a random sample of the inputs and then discarding values that do not fit the model. I have not yet tried either of these methods. The extraction method described below is a first step; it is simple and works with ideal data, but is not robust.

The extraction problem has been simplified by assuming that the environment is completely rigid and that the motion of cluster consists of a translation and three rotations.

This motion can be represented as a 4×4 matrix in homogeneous coordinates, just as described in standard graphics texts (e.g. [Newman and Sproull, 1979]).

$$M = M_{\theta_{az}} M_{\theta_{el}} M_{\theta_{rl}} M_{T_{xyz}} \text{ where}$$

$$M_{\theta_{az}} = \begin{pmatrix} \cos \theta_{az} & -\sin \theta_{az} & 0 & 0 \\ \sin \theta_{az} & \cos \theta_{az} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$M_{\theta_{el}} = \begin{pmatrix} \cos \theta_{el} & 0 & \sin \theta_{el} & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_{el} & 0 & \cos \theta_{el} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$M_{\theta_{rl}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_{rl} & -\sin \theta_{rl} & 0 \\ 0 & \sin \theta_{rl} & \cos \theta_{rl} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \text{ and}$$

$$M_{T_{xyz}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}.$$

The shift (S_i) reported by a sensor chip (i) can be represented in homogeneous coordinates as $O_i = (S_i \ 0 \ 0 \ 1) W$, where W is a scale factor. This row vector is related to the known range to the scene (R_i), the known viewing matrix for the sensor (V_i), and the unknown motion matrix for the cluster (M) by

$$O_i = (0 \ 0 \ R_i \ 1) V_i^{-1} M V_i. \quad (11)$$

This expression can be expanded to produce a non-linear equation in 6 unknowns, three rotations (θ_{az} , θ_{el} , θ_{rl}), and three translations (T_x , T_y , T_z). The equation for each sensor is in these same 6 unknowns so we have a set of simultaneous non-linear equations.

3.3.2 Solution Methods

Non-linear. The solution could be directly determined by solving the system with an algorithm designed for non-linear simultaneous equations, for example multidimensional Newton's method [Acton, 1970], or Brown's nonlinear extension of Gauss reduction [Brown, 1973]. This approach has two serious difficulties. The first is uniqueness of the solution; systems of non-linear equations can have many solutions. The second is efficiency. Solution of this system of non-linear equations may require many iterations to converge and each iteration will require evaluation of sines and cosines. Current super-microcomputers or even super-minicomputers would be incapable of solving the system in 1 millisecond and the problem is not sufficiently regular to allow efficient solution on an attached array processor.

Linear Approximation. A much more efficient solution method results from converting the non-linear system of equations to a linear system that has approximately the same solution. One such approximation is based on eliminating the sines and cosines by using the small angle approximations, $\sin \theta = \theta$ and $\cos \theta = 1$, and ignoring products of small angles. Simulations (section 3.4) indicate that these simplifications are made valid by the high frame rate of the sensors relative to the speed of human motion—yet another simplification resulting from a high frame rate. The resulting motion matrix is

$$M = \begin{pmatrix} 1 & -\theta_{az} & \theta_{rl} & 0 \\ \theta_{az} & 1 & -\theta_{el} & 0 \\ -\theta_{rl} & \theta_{el} & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix} \quad (12)$$

Substituting this into equation (11) we obtain a linear system of equations in six unknowns, with one equation for each sensor in the SELF-TRACKER cluster. The least-squares solution to this system can be efficiently determined using standard methods. I plan to use this solution directly, but it could be used as a starting point for a fast iterative improvement procedure to obtain an exact solution (e.g. [Hirvonen, 1971]).

3.4 A Simulation Study of Accumulated Error

The bounds analysis in section 3.1 gives bounds on how badly the system may perform; it says nothing about error cancellation resulting from multiple redundant sensors. To get a better feel for how errors will accumulate in an actual system I programmed a simulation of a cluster moving around in a room. The simulation program allows specification of the size of the room, the motion of the cluster, and all the parameters of the cluster (size, number of sensors, focal length, base-line, and pixel spacing). I assumed that the room is completely rigid (a approximation, since other people, or the user's hands or body, might be moving), that every point in the room has sufficient features to allow image registration, and that image registration is accurate to the nearest pixel. The program simulates the error in measurements of translation, range, and rotation, and also the error due to the linear approximation used in motion extraction, but it does not include the effects of image blur. The accumulated error was 6.3 cm of translation and 0.4 degrees of rotation after a simulated trip around a circle 1 meter in diameter at 1 meter per second with a rotation of 360 degrees. This error resulted in about 6 pixels of displacement in a 512×512 , 90 degree field-of-view head-mounted display. The parameters of the sensor cluster were the same as used throughout this chapter (400 pixels on a $13.5 \mu\text{m}$ pitch, 45 mm lenses with 50 mm base-line) with 10 sensors (5 stereo pairs). The simulated room was $4 \times 4 \times 2.5$ meters. This simulated 3.1 seconds of operation at 1000 frames per second.

After over 3 seconds of operation 6 pixels of error have been accumulated in the displayed image but the user has moved through a large distance (3.1 meters) and a large rotation (360 degrees). During this time a combined system would have gotten several fixes on beacons, thus reducing the accumulated error.

Chapter 4

A Chip for Natural-Environment Tracking

4.1 Motivations for Implementation

My primary motivation for implementing the prototype is that it is a necessary step toward achieving a working system. The goal of my work is more than a paper design; I want a working system for use with a head-mounted display system.

Another important motivation is to demonstrate the validity of the basic assumptions. In chapter 3, the design decisions were based on the chip being fast enough, sensitive enough, and accurate enough. But is such a chip realizable? Practical?

Another motivation for implementing the prototype is to discover unanticipated problems with the concept or with the proposed implementation. Will images really register to within one pixel? Will noise generated by the digital signals in the processor degrade the performance of the imager?

A fourth motivation is to gain experience with the aggressive design style necessary to achieve systems of SELF-TRACKER's complexity on a single chip. My design includes both analog and digital circuitry, synchronous and asynchronous control, and bit-serial and massively parallel computation.

4.2 Description of Chip

The SELF-TRACKER 1.0 chip described here is the fifth chip I have had fabricated as part of this research. The first four chips were tests of different photosensor designs (see chapter 5). A sixth chip, SELF-TRACKER 1.1, that corrects the timing problem described later went to MOSIS on February 20, 1984.

The SELF-TRACKER 1.0 chip consists at the highest level of three major sections: imager, processor, and control. The position and sizes of these three sections are shown in Figure 4.1. The function of the photosensor array is to form a bi-level (bit) representation of the image that will be focused onto it by a lens. The bi-level image is shifted serially from the photosensor array into the upper right corner of the processor. The processor compares the new image from the photosensor array with a previous image held in one

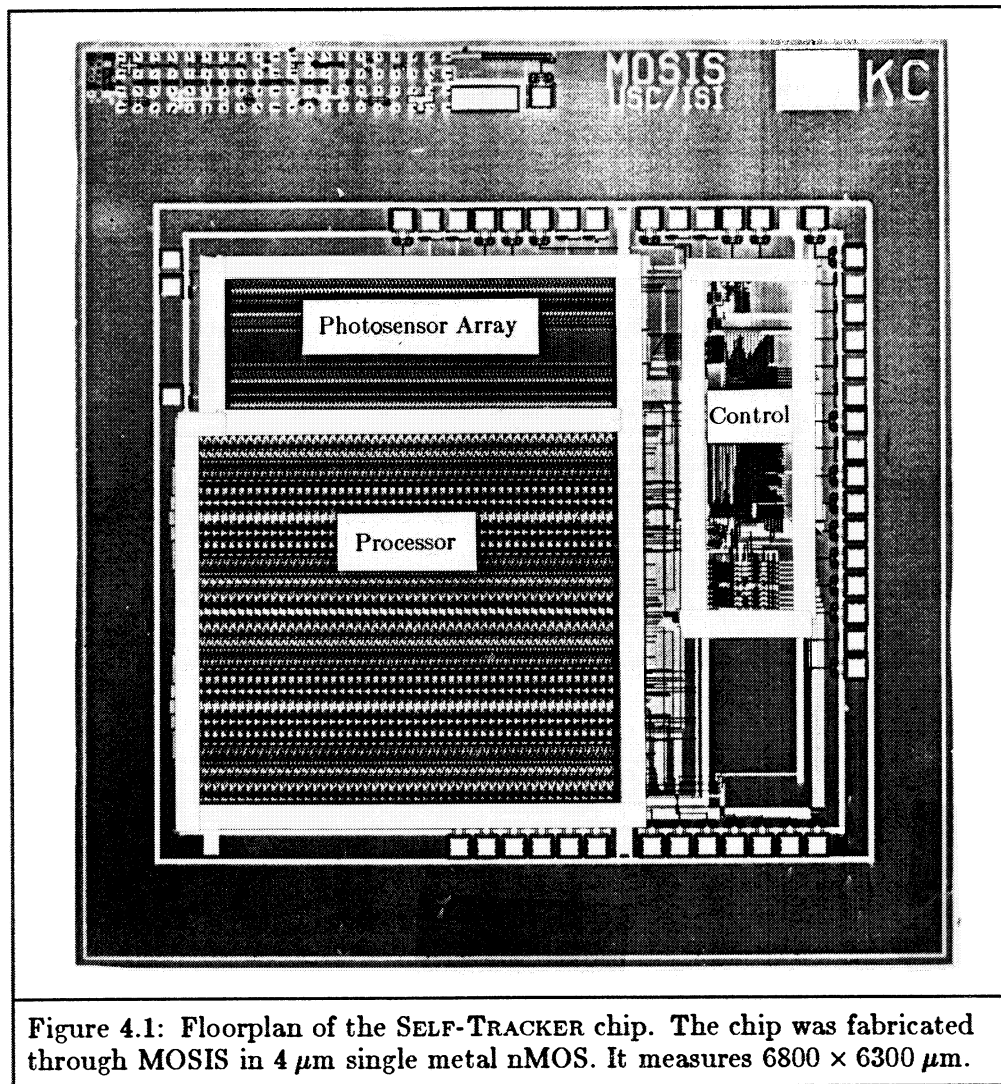
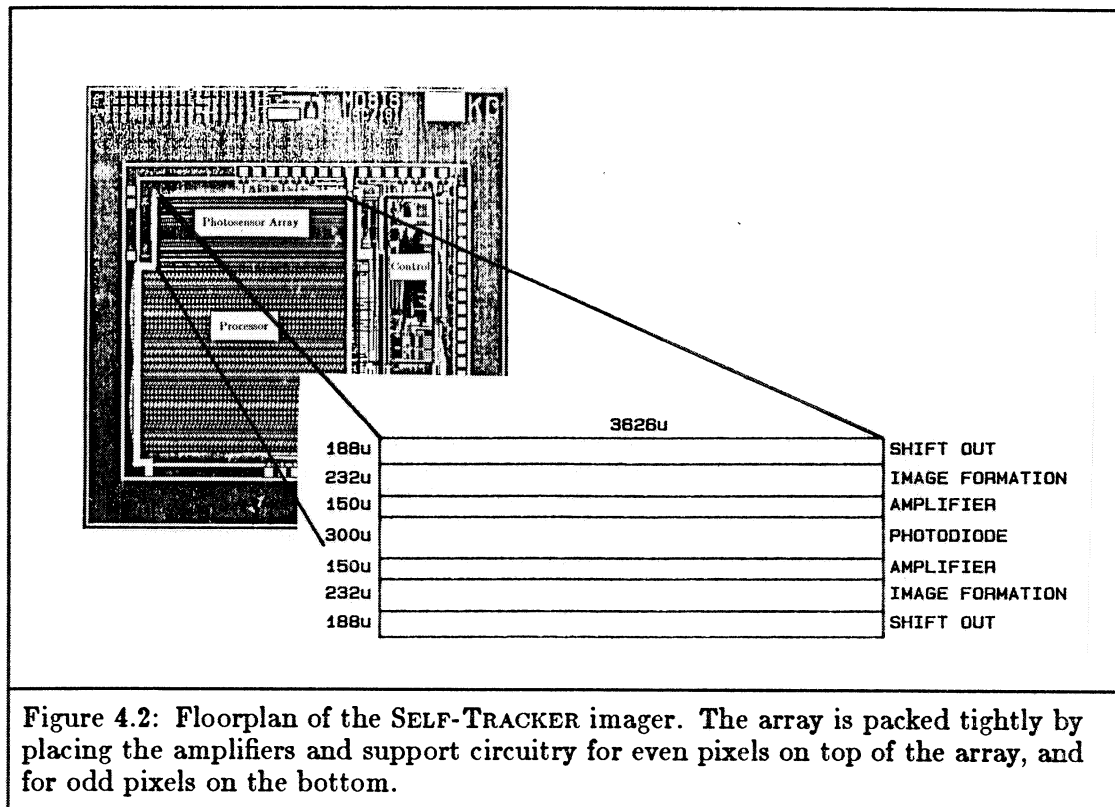


Figure 4.1: Floorplan of the SELF-TRACKER chip. The chip was fabricated through MOSIS in $4\text{ }\mu\text{m}$ single metal nMOS. It measures $6800 \times 6300\text{ }\mu\text{m}$.

of its registers to determine the amount image shift. The control section generates the clock and control signals for the photosensor array and the processor. I will describe the implementation of each of these sections separately.

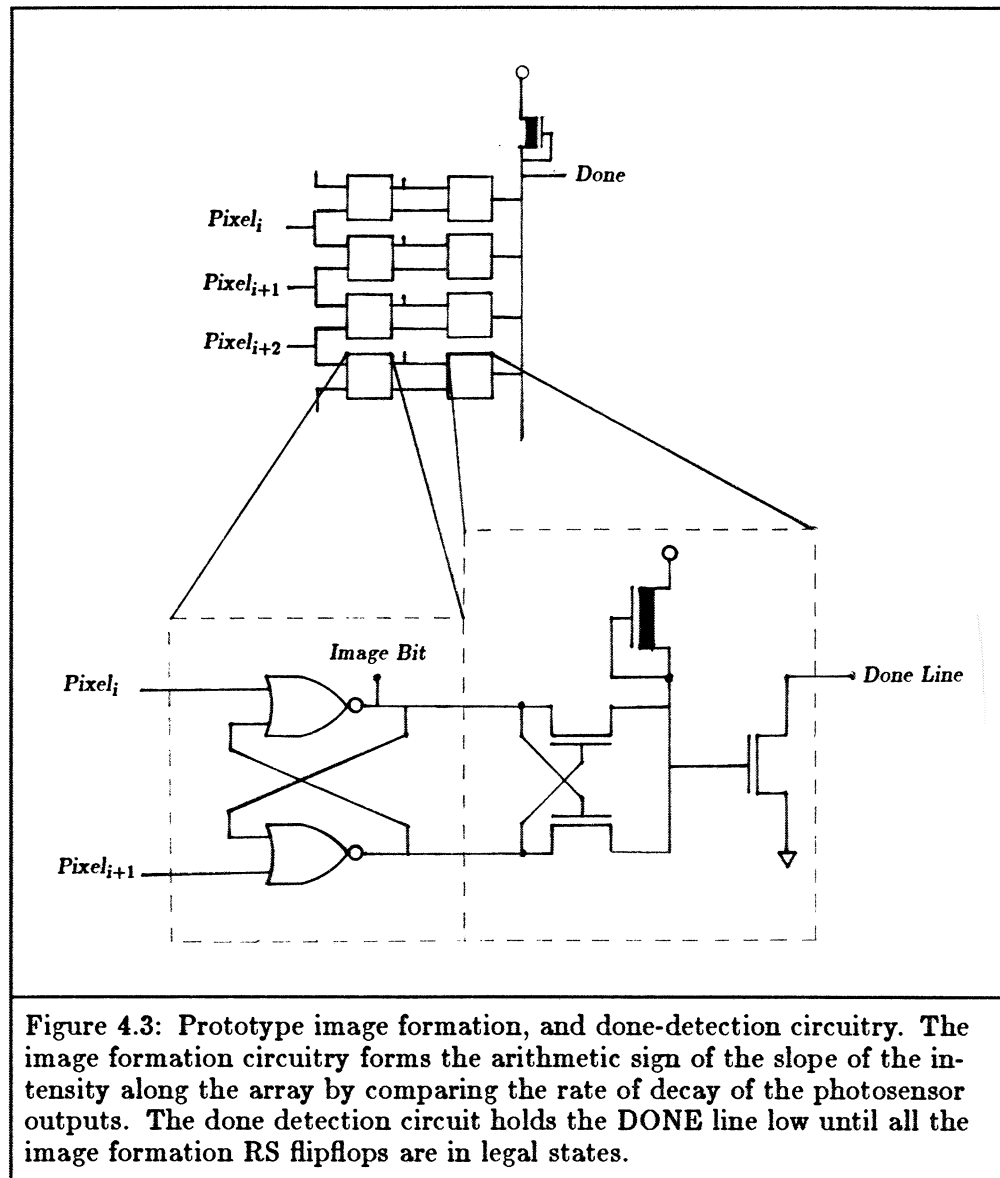
4.2.1 Imager

Figure 4.2 shows the floorplan of the imager. The light is collected by the photosensors, amplified by the amplifier array, converted to bi-level by the image formation circuitry, and latched and shifted out by the shift register. The array consists of 200 cells on a $36\text{ }\mu\text{m}$ pitch. The photosensor array is interdigitated with even pixels on top and odd pixels on bottom to allow pixels to be spaced as closely as possible. With $4\text{ }\mu\text{m}$ features the photosensors are on a $18\text{ }\mu\text{m}$ pitch; with $3\text{ }\mu\text{m}$ features I could achieve $13.5\text{ }\mu\text{m}$ pitch. The



design of the photosensors is described in chapter 5. The photosensors in the prototype are $12\ \mu\text{m}$ wide and $300\ \mu\text{m}$ long giving a sensor area of $3600(\mu\text{m})^2$.

Image Formation. The image formation circuitry used in the prototype is shown in the left box of Figure 4.3. The output is the arithmetic sign of the slope of the intensity along the array (section 3.2.2). The circuit is a RS flipflop driven, by the output of adjacent pixels. Both inputs start at 5 volts, driving both outputs low. The voltage at the inputs gradually drops to zero at a rate dependent on the incident light. Eventually one of the two inputs will differ from the other by enough to allow the feedback from the cross-coupled NOR gates to take over and force it into one of its two stable states. If the left pixel has more incident light, its output will go to zero faster and the value of the IMAGE BIT line will be one; if the right pixel has more incident light, the value will be zero. This circuit was chosen because it is simple and small, and because this approximation to edge detection was shown to be effective by the experiments in chapter 3.



Done Detection. This implementation determines that the image is completely formed by detecting the stable state of the RS flipflops used for image formation. The circuit is an analog XNOR gate described by Seitz in Mead and Conway, 1980 (the right box in Figure 4.3). It is connected to the Q and \bar{Q} outputs of the RS flipflop and produces a logical one when the output is unstable (Q and \bar{Q} are within one threshold) and zero when it is stable. The output of each pixel's XNOR gate is connected to a distinct input of a NOR gate. When all the RS flipflops are stable (all the output bits are decided), the output of the NOR gate goes to one, signaling that the image is ready for processing.

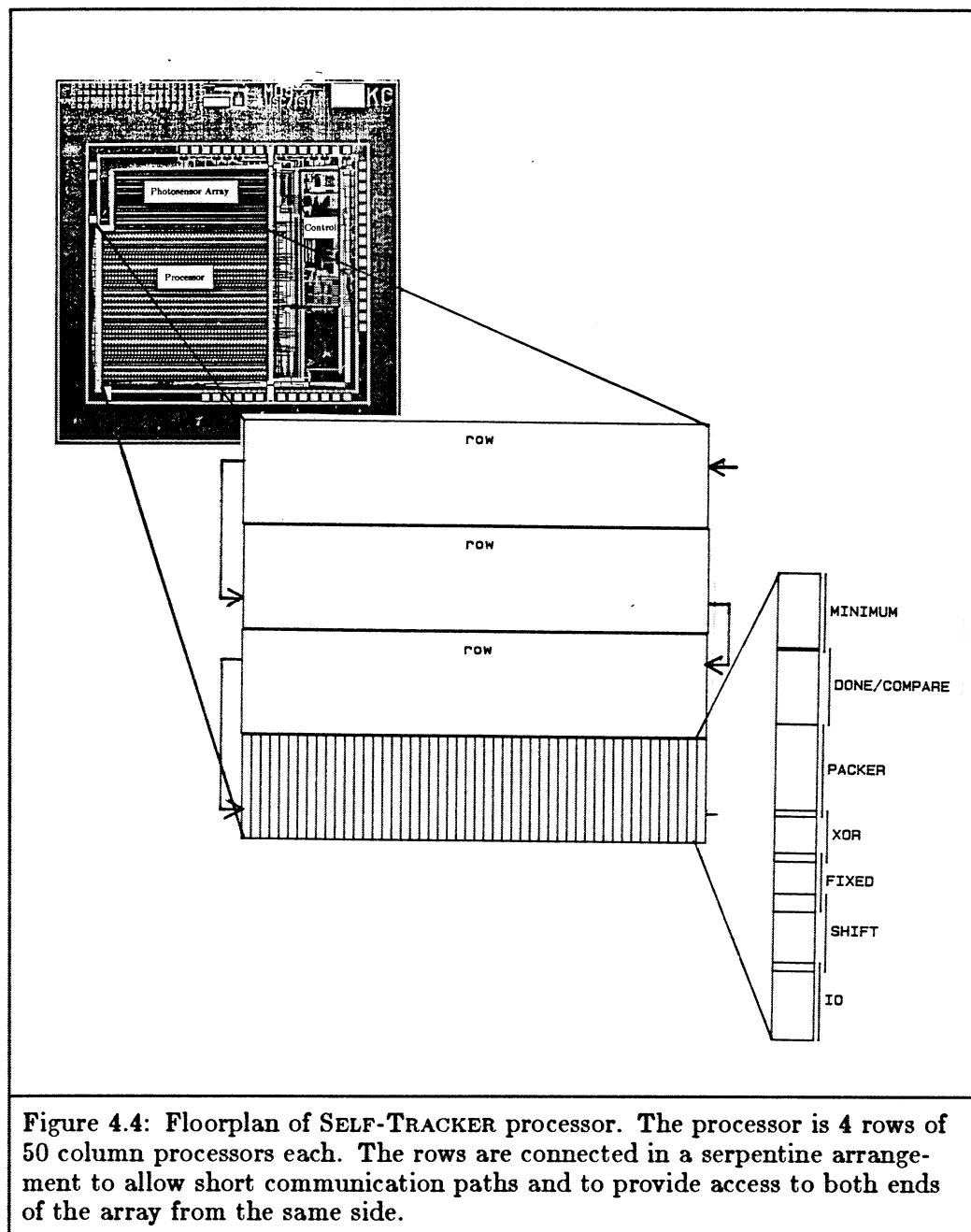
This circuit was chosen for the prototype because it is simple and because it worked well with the simulated images from the television camera. Two problems with the design were discovered when the chips were tested. The first problem was poor yield for the imager. With this design, any failure in the large NOR gate, or failure of two or more consecutive pixels to switch, will disable the done-detection circuit and thus the imager and the entire chip. The second problem is caused by small differences in the sensitivity of adjacent pixels. Because the circuit waits for all of the “races” to be won, some bi-level pattern will be formed even for perfectly uniform lighting (for example complete darkness). This pattern is the same for all image regions of constant intensity and causes the later registration process to be heavily biased towards zero shift. Both SELF-TRACKER 1.0 and 1.1 include this circuit. I will design a improved circuit for use in the SELF-TRACKER 2.0 chip planned for the fall of 1984. (see chapter 6.)

An improved done-detection circuit might wait for some portion of the bits to be decided and force all the undecided ones to zero. This could solve both the yield problem, by eliminating dependence on every pixel working, and the noise problem, by eliminating decisions that are too close to call. I haven’t investigated this solution yet, but it should be easy to implement using a “half-done” circuit similar to Tanner’s [Tanner and Mead, 1984], and it should also be easy to simulate by multiplying the individual pixels in the simulated images by slightly different values to simulate gain variations in the sensor array.

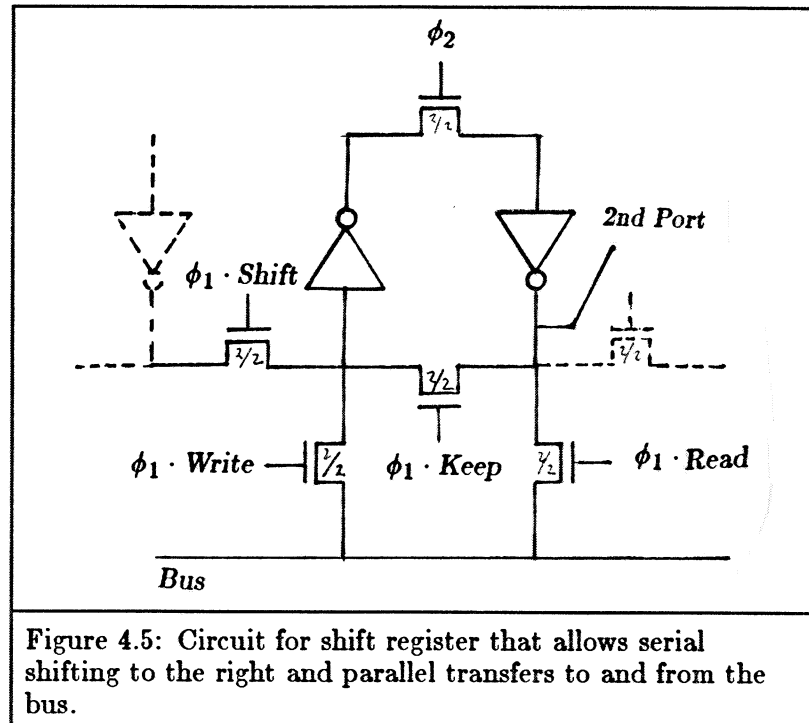
Provisions for Test. Testing the imager is facilitated by the external control provided by the DECODER, described later, and by access to the input and output of the serial shift register from external pins. I first tested the shift register by serially transferring bit patterns from its input to its output. I then tested the array by resetting, waiting for DONE, latching the image, and shifting it out to the test computer.

4.2.2 Processor

The floorplan of the processor is shown in Figure 4.4. The 4 rows of 50 columns are each arranged as a serpentine to allow the processor to fit in to an almost square region. This arrangement also provides input and output for serial transfers on the same side of the array. The design element here is the column which is a bit-slice of the 200-bit processor. The column consists of four registers (IO, SHIFT, FIXED, and MINIMUM), an exclusive-or gate, and an asynchronous circuit that simplifies the image disparity comparison. Three of the registers (IO, SHIFT, and MINIMUM) provide for shift-right operation as well as parallel transfers. The IO register is used for bit-serial communication with the photosensor array and for testing. The SHIFT register holds



the moving image during the image comparison operation. The **FIXED** register holds the stationary image. The **XOR** gate compares the bit in **SHIFT** to the corresponding bit in **FIXED**. **PACKER** synchronously latches the disparity value from the **XOR** gate and asynchronously converts it to an unary representation of the amount of disparity by moving all the 1 bits to the right and all the 0 bits to the left. The resulting unary representation is compared to the current minimum (in **MINIMUM**) by the **DONE/COMPARE** circuit. The



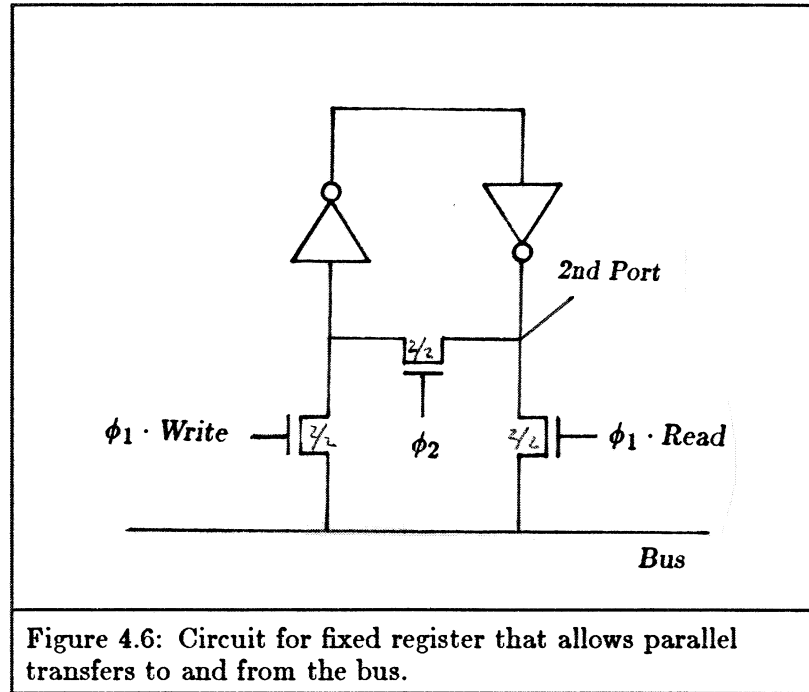
shift measurement algorithm is implemented from these primitive operations as described in section 4.2.3.

Register Design. The shift registers in the processor are pseudo-static and allow for parallel and serial transfers. The circuit is shown in Figure 4.5.

The fixed register, Figure 4.6, is basically the same as the shift register with the control signals and transistors for shift operation removed.

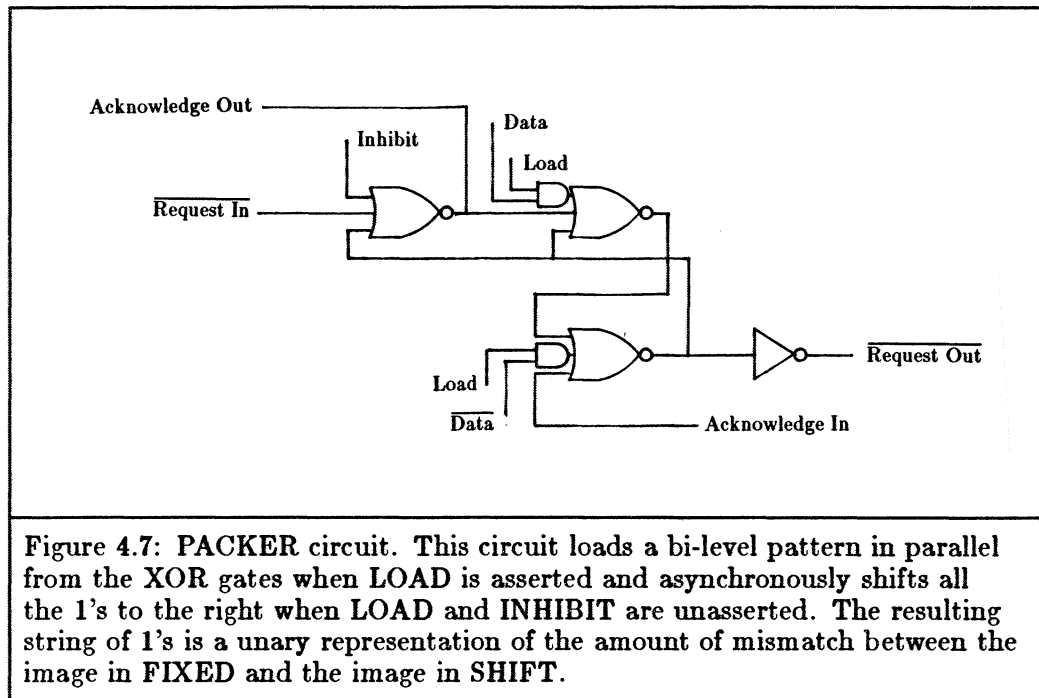
These two cells are used for the IO, SHIFT, FIXED, and MINIMUM registers in the column. SHIFT and FIXED are made dual-ported by adding a connection to the points labeled “2nd Port” in Figure 4.5 and Figure 4.6. Together they make up 55% of the area of the processor and 20% of the entire chip.

Comparing Images. The images in SHIFT and FIXED are compared using one exclusive-or gate for each bit to produce logical one when the bits in SHIFT and FIXED are different and zero when they are the same. The inputs of the XOR come from the second port of SHIFT and FIXED; the output drives the input of the PACKER.



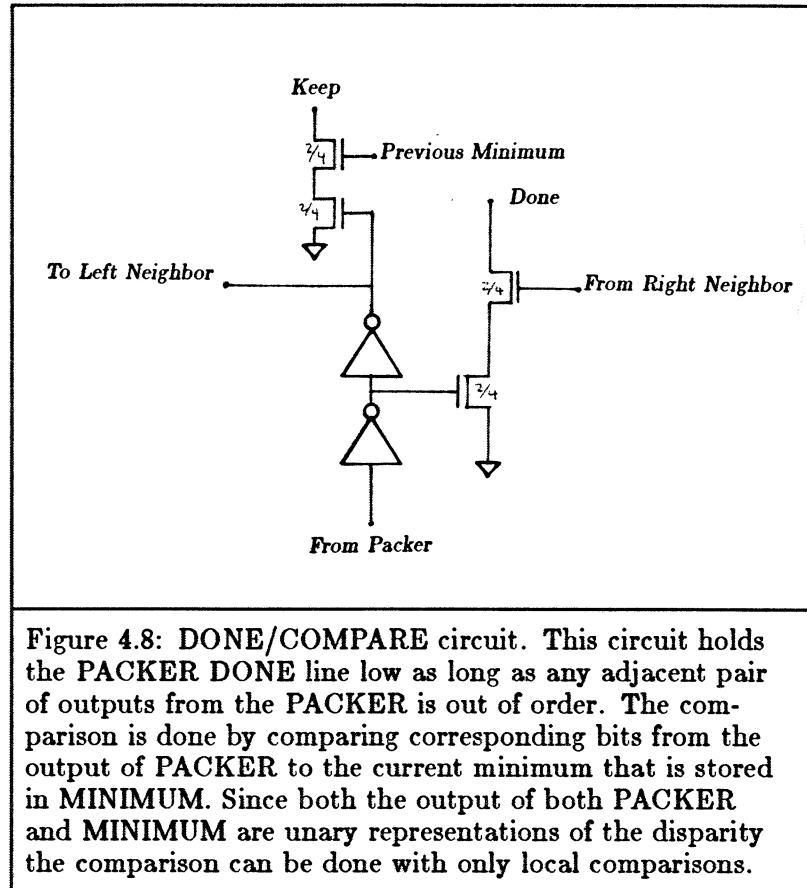
Comparing Disparity. Finding the shift that minimizes the difference between two images requires some method of comparing the output of the XOR gates at one relative shift to that at another shift to determine which has fewer ones. The time constraints are severe; 1000 times per second the number of ones in two 200-bit strings must be compared for 64 different shifts. (This is for the SELF-TRACKER 1.0 chip. The final design will have 300 to 400 bit strings and will measure both motion shift, 64 positions, and range shift, 150 to 200 positions.) The obvious solution of counting the number of bits and comparing the counts is not practical because a serial adder would be too slow, a parallel adder would be too big, and an analog adder would not be accurate enough. A serial adder must operate at 13 million bits per second to count the bits. A parallel adder tree to sum the ones would consist of about 500 rather large adder cells. An analog approach, in which each 1 from the XOR gates would contribute a small current into a capacitor with the time to rise to a threshold voltage as the basis for comparison, would require that the current sources be matched within 1 part in 200 in order to detect one bit of difference in 200 bit vectors. I felt that this degree of matching would be too difficult to achieve with a standard nMOS process.

I solved the comparison problem with a circuit that converts an arbitrary string of ones and zeros into a string with all the ones at the right end and all the zeros at the left; a unary representation of the disparity. After conversion to this representation, two strings can be



compared with trivially simple circuitry to determine which string of ones is shorter and thus which shift has less disparity. The conversion to unary is handled by the asynchronous circuit shown in Figure 4.7. The output of the XOR gates is latched into the array of PACKER cells synchronously in parallel when the LOAD and INHIBIT signal are high. Then LOAD and INHIBIT go low (this causes a serious problem described later). Each PACKER cell that contains a one asserts the REQUESTOUT line to its right neighbor. Each PACKER cell that contains a zero and that has its REQUESTIN line asserted will change its state to one and assert its ACKNOWLEDGEOUT line. When a PACKER cell that contains one has its ACKNOWLEDGEIN line asserted, it resets to zero. Thus the ones shuffle concurrently to the right, as fast as possible. The DONE/COMPARE circuit shown in Figure 4.8 detects completion of the packing operation by holding the DONE line low as long as any adjacent pair of bits are out of order (the left bit is one and the right bit is zero). The comparison operation is done using the unary images by comparing corresponding bits in the current best match, held in MINIMUM, to the output of the PACKER.

The concurrency in the PACKER circuit greatly speeds its operation, but this design is fast enough even in the worst case of transferring a single bit from the far left to the far right. This circuit operates at about 16 nanoseconds per stage with $4\text{ }\mu\text{m}$ circuit



features, giving a worst-case time of 3.2 microseconds for a 200 bit processor; 64 shift comparisons at 1000 frames per seconds allows 16 microseconds per comparison. The next processor, with 300 bit images, will be implemented with $3\ \mu\text{m}$ circuit features. I estimate that the PACKER will operate at 12 nanoseconds per stage, giving a worst case time of 3.6 microseconds. The 300 bit design will measure range as well as motion, requiring comparison at about 200 different positions and thus will allow 5 microseconds per comparison.

The problem, mentioned earlier, when LOAD and INHIBIT go low, is a race condition that allows a PACKER cell to shift out a one without resetting to zero. This race was not discovered during simulations of the chip because the switch level simulator I used (esim), assumes synchronous operation. The problem manifests itself in the prototype chip by producing many more ones in the output than were in the input pattern when consecutive ones are loaded into the first or third rows of the PACKER. The second and fourth rows work as expected and the third row will pass consecutive ones that were loaded into the second row. These strange characteristics and extensive simulations with SPICE 2G.5

lead me to the conclusion that the problem can be eliminated by delaying the INHIBIT signal with respect to LOAD. The delay has been implemented by driving the INHIBIT line with a noninverting superbuffer that is driven by the far end of the LOAD line, thus assuring that INHIBIT always falls to zero after LOAD. The delay has been implemented in the SELF-TRACKER 1.1 chip that went to MOSIS for fabrication on February 20.

Provisions for Test. The shift registers (IO, SHIFT, and MINIMUM) in the processor greatly simplify testing it. The input and output of each of these registers are brought out to pads so that along with the control signal generation facilities of the DECODER module, the registers and other circuitry can be tested with arbitrary bit patterns shifted in from our test computer. The test programs first tested the registers for serial transfers, then for parallel transfers and then tested the XOR, PACKER, and DONE/COMPARE circuitry using patterns shifted into the registers.

4.2.3 Control and Signal Generation

Figure 4.9 shows the floorplan of the control section of the SELF-TRACKER prototype. This section generates the signals that control the processor and implement the image-comparison algorithm. The control section is almost entirely implemented from standard cells and PLA's. Its design was much more relaxed than the other circuitry because there were no special speed or space requirements.

Control Section Implementation. The image-comparison algorithm given in Figure 4.10 is implemented in MAIN using a 35-state, finite state machine, an eight-bit standard-cell counter circuit, an eight-bit latch, and some simple logic to detect certain counter states. The eight-bit counter and latch allow great simplification of the finite state machine by providing for the simple process of looping for a predetermined number of iterations.

The DECODER is a PLA that takes four input lines from MAIN and five from input pads and generates the 21 control signals needed by the processor and imager. The four lines from MAIN encode the operations needed in the shift measurement algorithm. These lines are effective only when all the lines from the five input pads are low. The five lines from pads allow generation of all the operations used by MAIN and others that are useful for chip test. This proved to be a most useful provision for testing, as all processor and imager operations could be initiated independently of the control section.

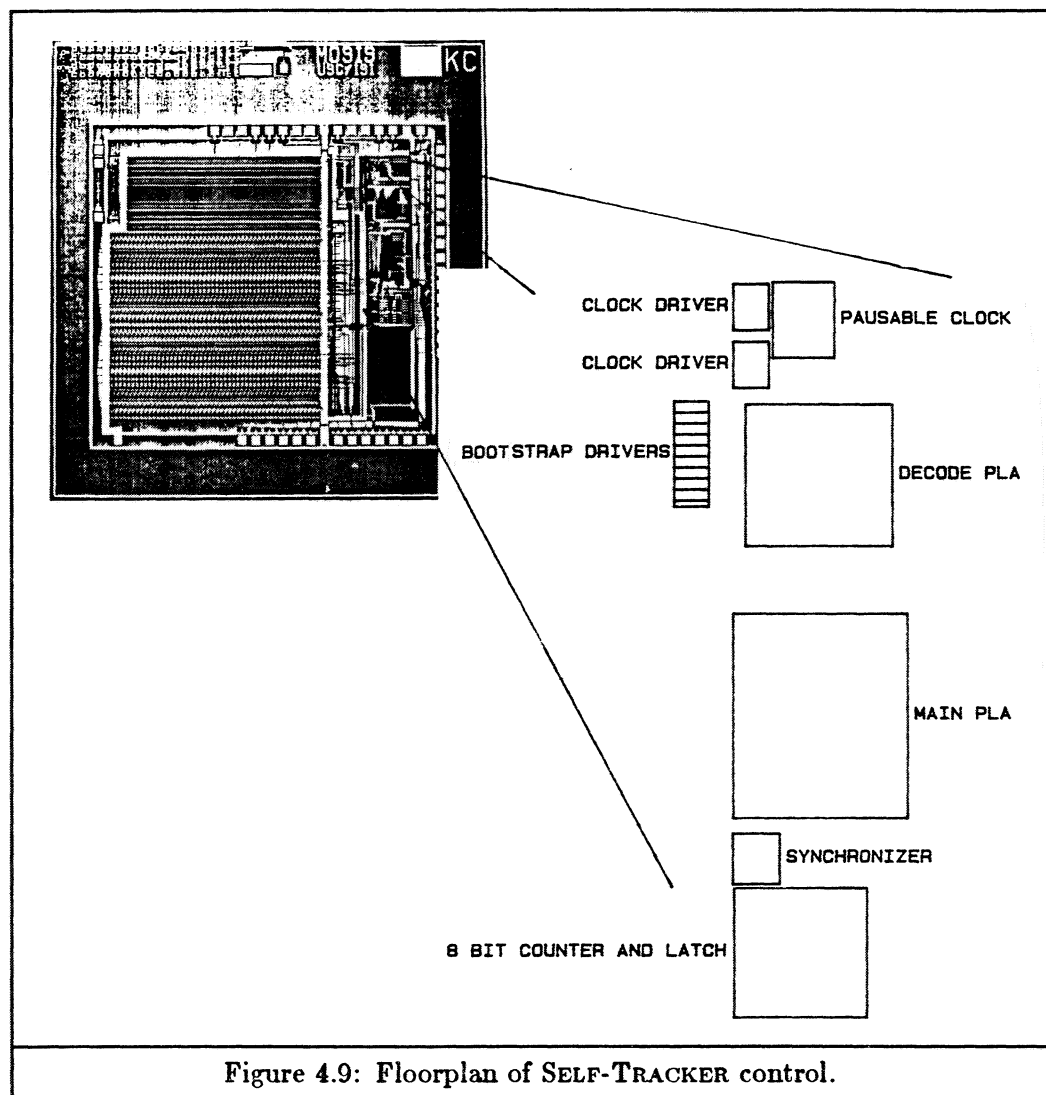


Figure 4.9: Floorplan of SELF-TRACKER control.

Clock Generation and Synchronization. A pausable clock is included in the prototype SELF-TRACKER to allow reliable synchronization of the asynchronous signals from the photosensor array and the DONE/COMPARE circuit. These signals are tested using a synchronizer circuit similar to the one described by Seitz in Chapter 7 of Mead and Conway, 1980. When the synchronizer enters a metastable state, the clock cycle is stretched to allow time for recovery.

Drivers. The drivers for control lines in the prototype use the bootstrap technique described to me by Seitz during a visit to UNC and described in Lutz *et al.*, 1984. The circuit, Figure 4.11, generates a qualified clock signal at full clock voltage and with considerable power with very small power dissipation. The driver does not amplify the

```

do forever
begin
wait until SENSOR-DONE is asserted
transfer the new image from the photosensor array into IO register
SHIFT := IO
PACKERIN := SHIFT xor FIXED
MINIMUM := PACKEROUT
count := 0
direction := left
latch := direction count
do
begin
shift SHIFT right
PACKERIN := SHIFT xor FIXED
if PACKEROUT < MINIMUM then
begin
MINIMUM := PACKEROUT
latch := direction count
end
count := count+1
end
until count = 32

SHIFT := FIXED
FIXED := IO
count := 0
direction := right
do
begin
shift SHIFT
PACKERIN := SHIFT xor FIXED
if PACKEROUT < MINIMUM then
begin
MINIMUM := PACKEROUT
latch := direction count
end
count := count+1
end
until count = 32

if count > 3 then
begin
report latch
count := 0
while MINIMUM-OUT = 1 do
begin
shift MINIMUM
count := count + 1
end
latch := count
report latch
end
end
end

```

Figure 4.10: SELF-TRACKER processing algorithm

clock, but rather passes it without significant attenuation whenever the control input is high.

The bootstrap line-drivers allow all the clock drive to be supplied in one place rather than distributed all over the chip. I was not able to provide this drive from off-chip as Seitz does, because this implementation needed a pausable clock for synchronization. The on-chip clock drivers switch 50pf of load capacitance at 2MHz, but they consume only 10

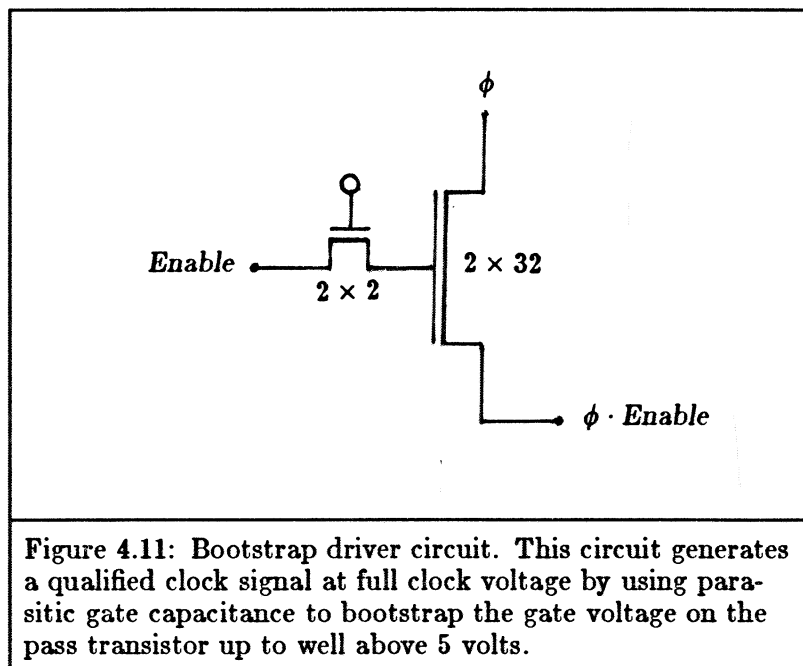


Figure 4.11: Bootstrap driver circuit. This circuit generates a qualified clock signal at full clock voltage by using parasitic gate capacitance to bootstrap the gate voltage on the pass transistor up to well above 5 volts.

milliwatts of static power. This low power consumption was achieved without sacrificing a full logic swing by using a small depletion transistor ($l/w = 4/100$) in parallel with a large enhancement transistor ($4/400$) as the pullup (Figure 4.12). The large enhancement transistor provides most of the drive without consuming static power, and the small depletion transistor provides the full logic swing. Figure 4.13 shows the result of a Spice simulation of the clock driver switching a 50pf load.

Provisions for Test. A major feature of the control section design is the ease of chip testing that it affords. The processor and imager can be easily separated from the control section for test through the five test inputs to DECODER. The MAIN PLA that implements the control algorithm can be controlled and monitored from pads, allowing verification of its operation. The clock generator can be disabled and the clock signal can be supplied from off chip, allowing simple interface to our test computer. The output of the clock drivers are available at output pads, allowing measurement of the internal clock frequency and verification of proper driver operation.

4.3 Testing Environment

Figure 4.14 shows the test head used for testing the SELF-TRACKER chips. The head consists of a 64 pin ZIF socket in a light-tight enclosure fitted with a "C-mount" for a standard television camera lens. The test head is connected to a PDP 11/23 system running UCSD Pascal. The PDP 11/23 is equipped with a 64 line parallel interface that

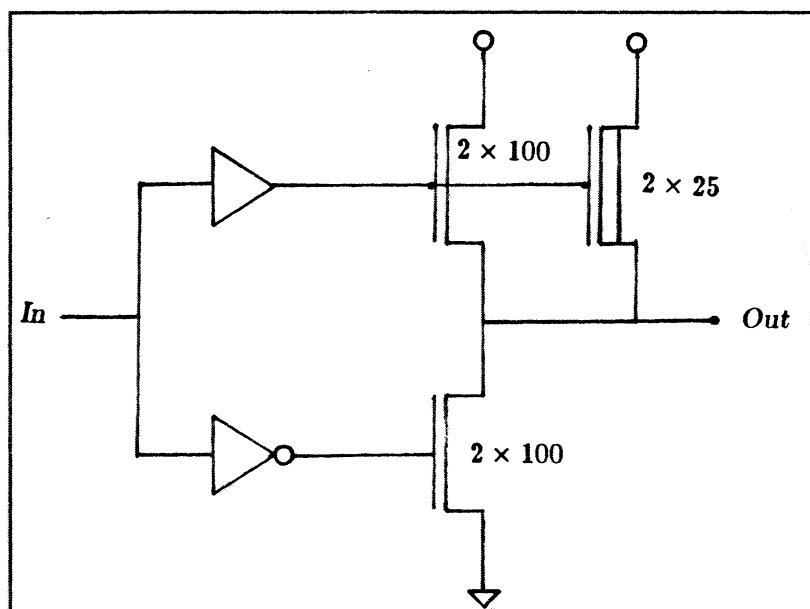


Figure 4.12: Clock driver circuit. The combination of a large enhancement-mode pullup to provide most of the drive and a small depletion-mode pullup to provide full logic levels produces a low power driver that can drive large loads.

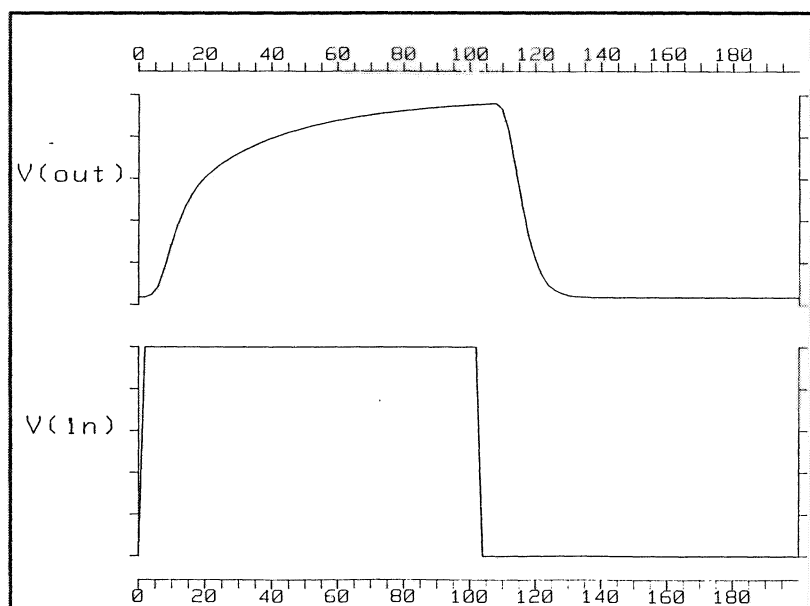


Figure 4.13: SPICE 2G.5 simulation of the clock driver switching 50pf.

has been modified to provide weak drive on its output lines. The weak drive is sufficient for input pads but is easily dominated by output pads.

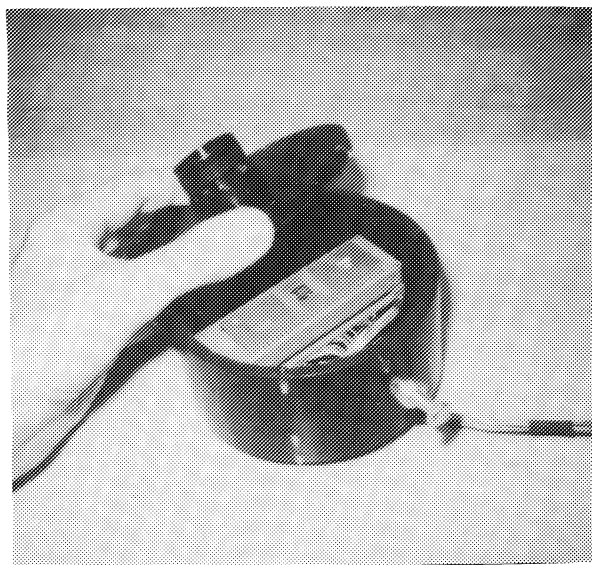


Figure 4.14: Camera test head for SELF-TRACKER chips

Chapter 5

Photosensor Design

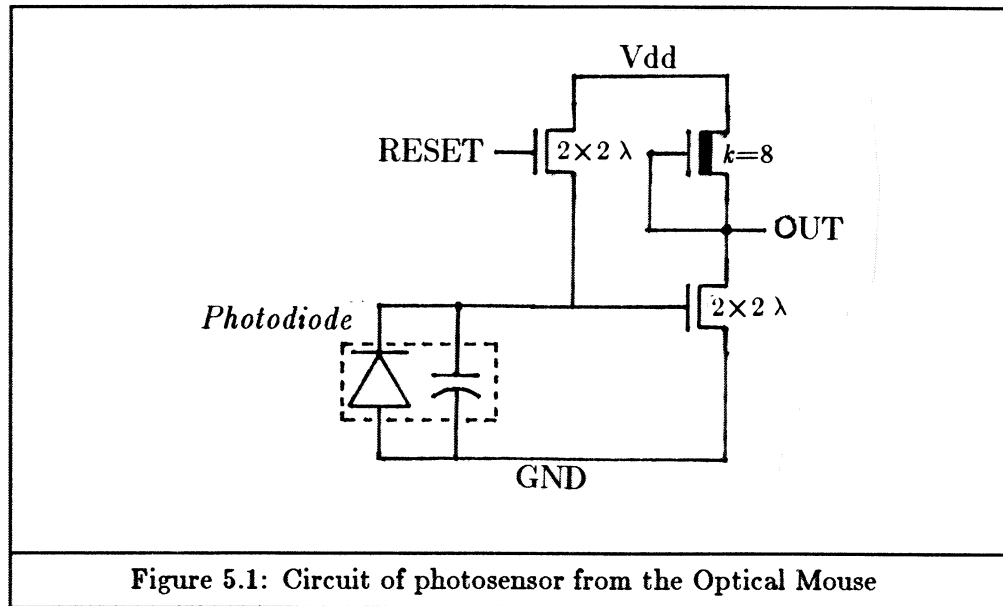
The design of sensitive photosensors that can be combined with digital circuitry on the same chip is the crucial problem in implementing a SELF-TRACKER. If the photosensors are not sufficiently sensitive, the system will be too slow to support the assumptions made in section 3.2.1. This chapter describes the design of photosensors that can be fabricated with standard nMOS processes and are sensitive enough for operation at thousands of frames per second under normal room light.

Photosensors in nMOS technology consist of a photodiode (an isolated region of diffusion) and an attached level-detection circuit. The photodiode is initialized by charging it to V_{dd} , thus driving electrons from the area. As photons strike the diffusion, they generate electron-hole pairs and the charge on the region is decreased (the absence of electrons is decreased) [Séquin and Tompsett, 1975; Howes and Morgan, 1979]. A level-detection circuit monitors the change in charge on the photodiode and switches at some fixed charge threshold. Since the amount of charge on the photodiode after reset, the conversion efficiency, and the level sensor threshold are constant, the time between reset and the signal from the level sensor is directly proportional to the flux of photons.

5.1 The Optical Mouse Photosensor

Richard Lyon's Optical Mouse [Lyon, 1981] uses photosensors such as the one in Figure 5.1. The photodiode is a square region of diffusion about $150\text{ }\mu\text{m}$ on each side and the level sensor is a series of three standard inverters, with pull up/pull down ratios (k) of 8, 4, and 8, respectively.

The first inverter switches its output from low to high when the voltage between the anode of the diode and ground crosses the threshold voltage of the inverter. This voltage is the product of the charge on the diode and its capacitance. Since the capacitance and the number of photons captured are both directly proportional to the area of the diode, the sensitivity of the sensor is dependent only on the threshold of the inverter.

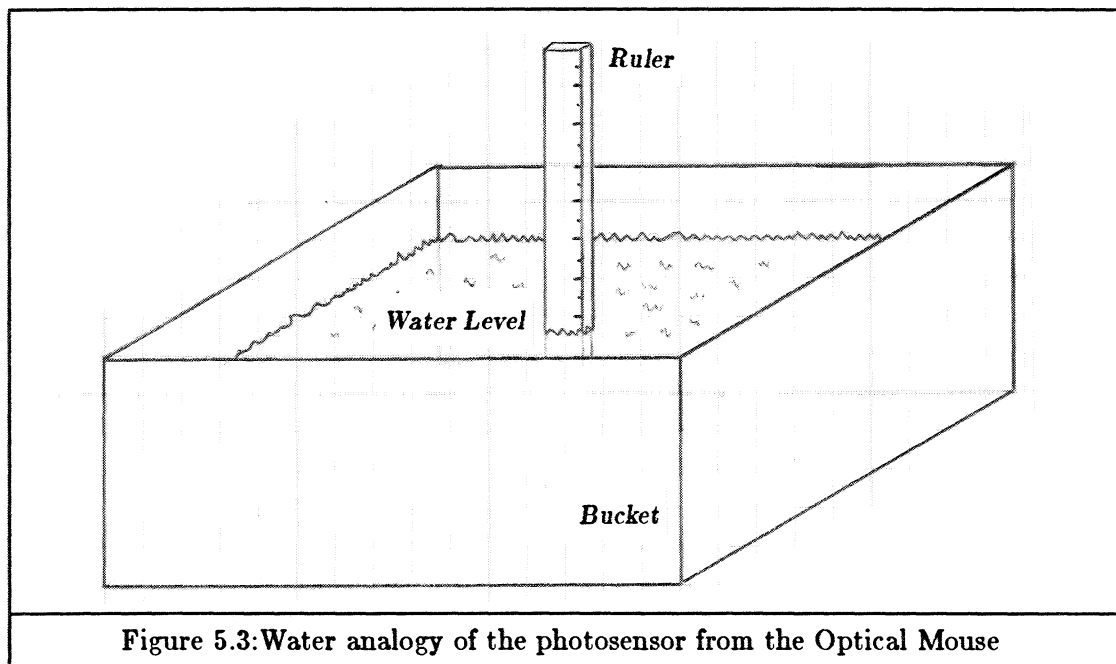
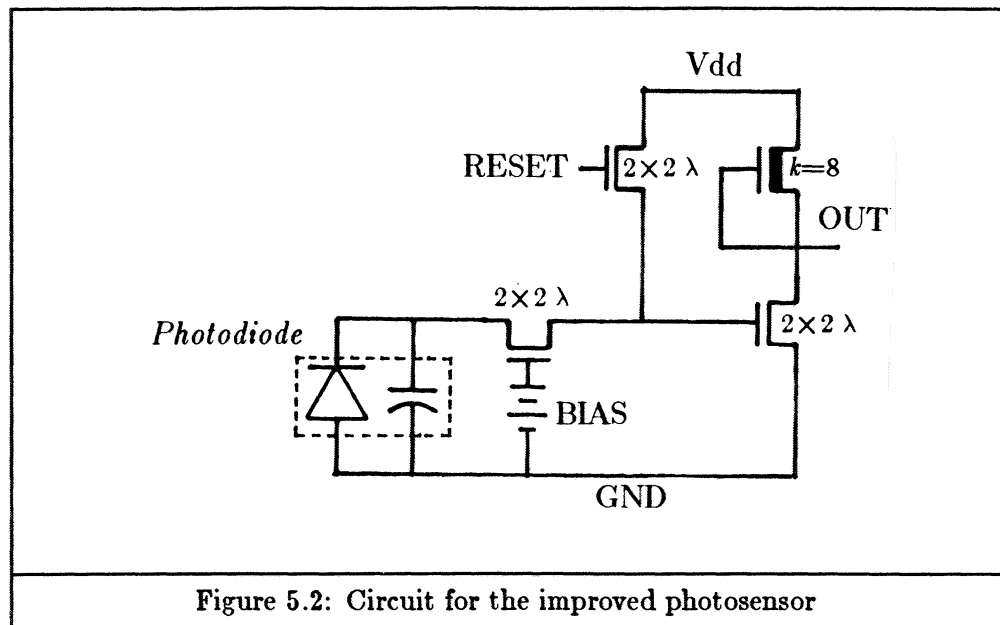


During the summer and fall of 1982, I designed, fabricated, and tested an integrated circuit that is a linear array of 200 copies of the Optical Mouse sensor except that my photodiodes were $14 \times 1000 \mu\text{m}$, about 60% as big. The sensitivity of the four chips I received was $32\text{nJ}/\text{cm}^2$ with a standard deviation of $2\text{nJ}/\text{cm}^2$. This sensitivity could be improved to about $18\text{nJ}/\text{cm}^2$ by making the photodiodes square (making the diode square reduces its sidewall capacitance).

Although this circuit was completely adequate for the Optical Mouse, even $18\text{nJ}/\text{cm}^2$ is not sensitive enough to allow millisecond operation in normal room light. It is not easy to improve the sensitivity of this design because it depends only on the threshold of the first inverter; the threshold of nMOS inverters is primarily determined by process parameters, not design parameters.

5.2 An Improved Photosensor

Carlo Séquin, during a visit to UNC in the fall of 1982, suggested an amplifier design with greatly improved sensitivity. The circuit is shown in Figure 5.2. A similar design is used in Herbst *et al.*, 1982. Its operation is most clearly explained using a water analogy suggested by Séquin. The basic problem is to measure the amount of rain (photons) falling into a bucket (photodiode). Lyon's approach (Figure 5.3) is analogous to placing a ruler in an empty bucket (the photodiode after reset) and sending a signal when the water level passes a fixed mark (the threshold of the inverter). The only way to improve the



sensitivity is to lower the mark on the ruler, since making the bucket bigger to catch more rain also makes its volume larger so that the water level does not rise any faster.

The improved design (Figure 5.4) uses a small bucket (small capacitor) to collect the runoff of water (charge) through a hole (barrier transistor) in the side of the bucket

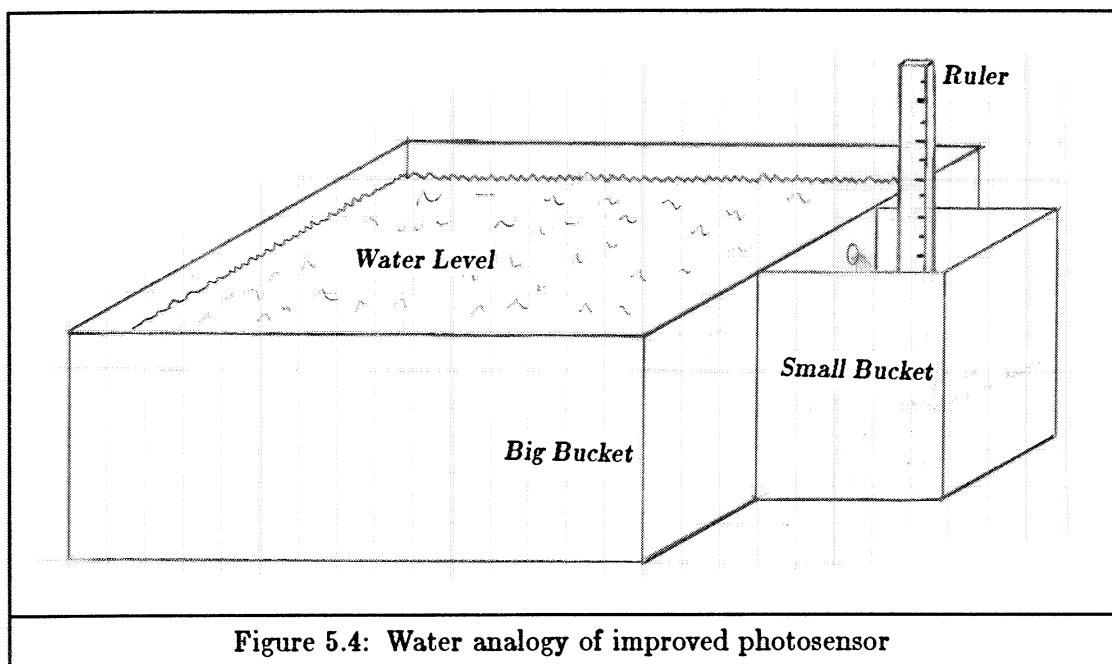


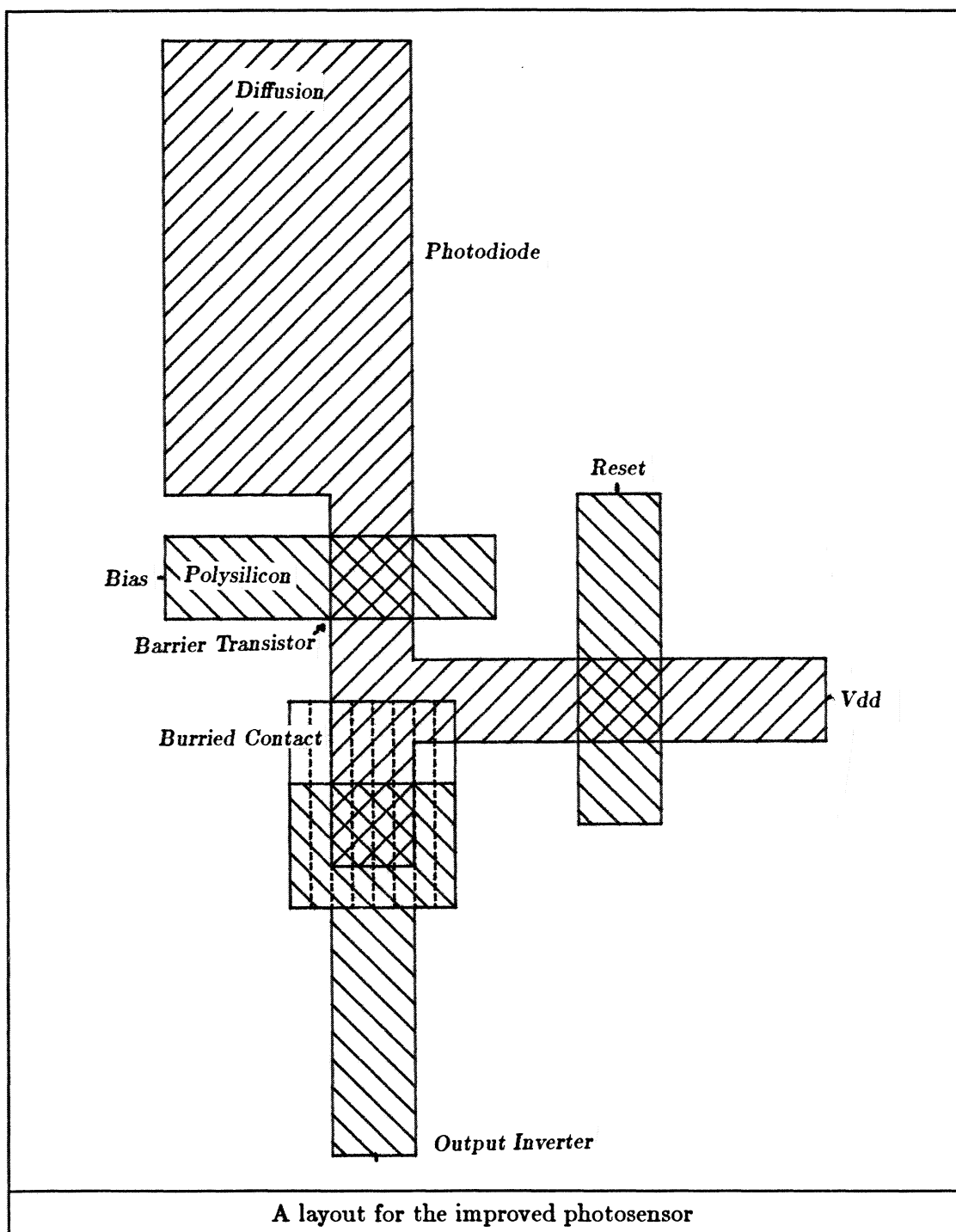
Figure 5.4: Water analogy of improved photosensor

(photodiode), near the top. Once the big bucket has filled to the level of the hole, all the rain that falls over its entire area will run off into the small bucket. By emptying (resetting to V_{dd}) the small bucket and waiting for the water level in it to rise to a fixed mark, the amount of rainfall can be measured in much less time. The improvement is, to a first approximation, the ratio of the size of the big bucket (the capacitance of the photodiode) to the size of the little bucket (the capacitance of a transistor gate).

5.2.1 Sensitivity of the Improved Photosensor

During the winter of 1983, I fabricated new chips with this improved amplifier design connected to photodiodes of three different sizes: $50 \times 50 \mu\text{m}$, $75 \times 75 \mu\text{m}$, and $200 \times 200 \mu\text{m}$. The measured sensitivity of these sensors is given in the following table. The measured sensitivity of even the smallest sensor is adequate for use with a natural-environment SELF-TRACKER if the optical gain of the lens is 20 or more. The optical gain is the ratio of the area of the lens to the area of the sensor.

Photodiode Size $\mu\text{m}/\text{side}$	Measured Sensitivity nJ/cm^2	On Chip Std dev nJ/cm^2	Chip to Chip Std dev nJ/cm^2	Frames per second at $2 \mu\text{mW}/\text{cm}^2$
50	1.88	0.07	0.27	1064
75	1.35	0.04	0.17	1471
200	0.46	0.02	0.05	4348

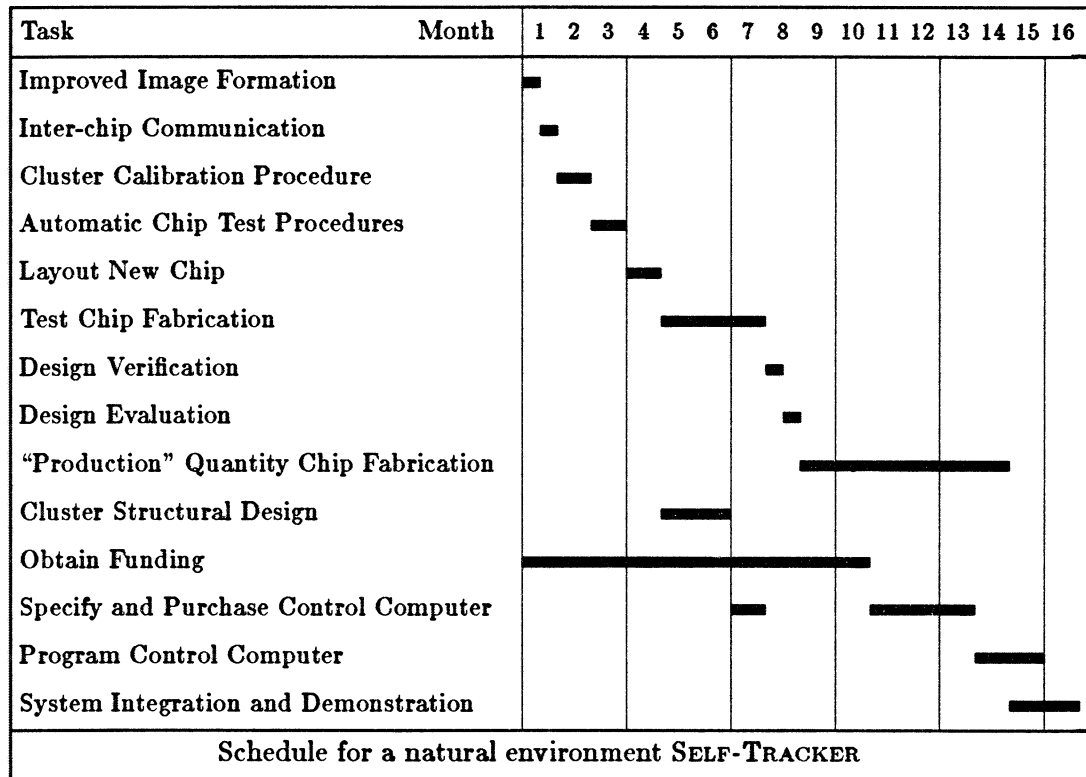


Chapter 6

Next Steps

6.1 Task List for a Natural Environment Tracker

This section lists the steps necessary to achieve a working natural-environment SELF-TRACKER. I propose to demonstrate a natural-environment SELF-TRACKER cluster in the fall of 1985.



6.1.1 Description of Tasks

Design Improved Image Formation. (1/2 month) As discussed in sections 3.2.2 and 4.2.1, a better method for forming bi-level images is needed. The current, sign-of-slope, method accentuates gain differences in the photosensor array. One possibility for a better method is to stop the current method after one-half the bits have been decided and set all the others to zero.

Design Inter-chip Communication. (1/2 month) An asynchronous communication protocol for transferring images from chip to chip and for communicating shift and confidence measurements to the host must be designed and implemented. I expect no difficulty since the data rates are relatively slow and the distances short.

Design Cluster Calibration Procedure. (1 month) The completed cluster must be calibrated; the design parameters probably cannot be used, as they were in the simulations, because of construction tolerances. The calibration procedure will probably require a simple “jig” that allows presentation of a controlled optical environment to the individual sensor chips in the cluster. The procedure should be designed before the final design of the sensor chips, because simple modifications to the photosensor array design may greatly simplify the calibration procedure.

Design Automatic Chip Test Procedures. (1 month) Test procedures must be designed that will allow the SELF-TRACKER 2.0 chips to be tested with minimum human effort. With expected yields, 50 to 100 chips will have to be tested to find the 10 functional ones that are needed for a working cluster. These procedures must be designed before final chip design since simple changes in the chip design can radically simplify testing.

Layout New Chip. (1 month) After the design of the improved image formation circuitry, the inter-chip communication circuitry, and the calibration procedure is complete, the natural-environment SELF-TRACKER 2.0 chip should require about 1 month. Almost all the circuitry from the current SELF-TRACKER 1.1 chip will be used in the new design.

Test Chip Fabrication. (3 months) The initial fabrication of the SELF-TRACKER 2.0 chip will probably require two to three months, based on our past experience with MOSIS. I hope this fabrication run will produce two or three working chips.

Design Verification. (1/2 month) After receiving the initial fabrication run of SELF-TRACKER 2.0 chips, the design will be verified using our current test head (see section 4.3). If at least two working chips are recieved from the first fabrication run, I will be able to test and demonstrate the complete cycle of operation for a pair of sensor chips, including measuring range and motion.

Design Evaluation. (1/2 month) If all goes well in the previous design verification, I will move toward fabricating a cluster. If some problems are found in the design, the schedule will be slipped at this point.

"Production" Quantity Chip Fabrication. (6 months) The natural-environment SELF-TRACKER cluster will require 10 working chips. If the yield on chips returned from MOSIS is 20%, 50 or more chips will have to be fabricated to get 10 that work. MOSIS is gearing up to provide "production" quantity volume, but I anticipate 4 to 6 months delay before sufficient chips are available.

Cluster Structural Design. (2 months) The design of the natural-environment SELF-TRACKER cluster will be complicated by the need to get 10 sensor chips with wiring and optics into a relatively small space. This design can occur in parallel with the preceding activities. Funding becomes critical at this point to pay for these services.

Obtain Funding. (10 months) The later stages of this project require substantial funding to purchase needed equipment and services. A research proposal to NSF and DARPA will be written during the first month. The remainder of the allotted time is the typical delay before funding.

Specify and Purchase Control Computer. (4 months) The specification of the control computer can occur while waiting for funding. I believe that a super-microcomputer comparable to the MASSCOMP 500 with an attached array processor and options for real-time data collection will be adequate. After funding is available, purchasing will require two to three months.

Program Control Computer. (2 months) The control computer must be programmed to collect the data from the chips in the cluster and to extract the three-dimensional motion of the cluster from the shift data reported by the chips. Both of these programming tasks are complicated by the real-time nature of the problem.

System Integration and Demonstration. (2 months) System integration includes debugging the hardware and software for communicating with the cluster, calibration of the cluster, and intial system checkout.

6.2 Further Research for a Combined System

The next step toward a usable SELF-TRACKER system must be some method of conquering the drift inherent in the natural-environment SELF-TRACKER. I believe that the system with both beacon and natural-environment tracking (chapter 2) is the most promising solution. The next step then should be to undertake development of a beacon-based system as a largely independent research project.

Another important area for further research is a robust method for extracting the cluster's motions from the shifts reported by the sensor chips. The method that I propose in chapter 3 may be satisfactory for a demonstration, but in a practical system some method that is more tolerant to erroneous reports from the sensor chips will be necessary. I plan to investigate optimal estimation and RANSAC [Fischler and Bolles, 1981] as the basis for an improved motion-extraction algorithm.

A final area for further research is a method for combining the measurements from a natural-environment system with those from a beacon system to provide a better estimate of the cluster's three-dimensional motion. The sightings from the beacon sensors will often not include enough beacons to allow absolute determination of the cluster's position. Some method must be found that uses knowledge of the cluster's current position, output of the motion sensors, and whatever beacon sightings are available to produce a best estimate of the clusters position.

References

- Acton, F. S. 1970. *Numerical methods that work*, Harper and Row, New York, NY.
- Ballard, D. H. and C. M. Brown. 1982. *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ.
- Bishop, G. 1982. *Gary's Ikonas Assembler, Version 2; Differences Between Gia2 and C*, TR82-010 UNC Department of Computer Science, Chapel Hill, NC.
- Brown, K. M. 1973. "Computer oriented algorithms for solving systems of simultaneous nonlinear algebraic equations," *Numerical solution of systems of nonlinear algebraic equations*, G.D. Byrne and C.A. Hall (eds.), Academic Press, New York, NY.
- Burton, R. P. 1973. *Real-time measurement of multiple three-dimensional positions*, Ph.D. dissertation, Department of Computer Science, University of Utah, Salt Lake City, UT.
- Clark, J. H. 1976. "Designing Surfaces in 3-D," *Communications of the ACM*, Vol. 19, No. 8, 454-460.
- Fischler, M. A. and R. C. Bolles. June 1981. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, 381-395.
- Fuchs, H., J. Duran, and B. Johnson. 1977. "A system for automatic acquisition of three-dimensional data," *AFIPS Conference Proceedings*, Vol. 46, 49-53.
- Gelb, A. 1974. *Applied Optimal Estimation*, MIT Press, Cambridge, MA.
- Herbst, H., H. Grassl, and H. Pfeleiderer. 1982. "Experimental Autofocus System for Lens Shutter Cameras," *IEEE Journal of Solid-State Circuits*, Vol. SC-17, 558-561.
- Hirvonen, R. A. 1971. *Adjustment by Least Squares in Geodesy and Photogrammetry*, Frederick Unger Publishing, New York, NY.
- Howes, M. J. and D. V. Morgan. 1979. *Charge-coupled Devices and Systems*, Wiley, New York, NY.
- Kilpatrick, P. J. 1976. *The use of a kinesthetic supplement in an interactive graphics system*, Ph.D. dissertation, Department of Computer Science, UNC, Chapel Hill, NC.
- Landman, H. A. September 1983. "OPUSI: An Optical Digital Position Sensor," *IC-CAD 83*.
- Liebelt, P. B. 1967. *An Introduction to Optimal Estimation*, Addison-Wesley, Reading, MA.

- Lutz, C., S. Rabin, C. Seitz, and D. Speck. January 23, 1984. "Design of the Mosaic Element," *Proceedings, Conference on Advanced Research in VLSI, MIT*, Paul Penfield, Jr. (ed.), Artech House, Dedham, MA, 1-10.
- Lyon, R. F. 1981. "The optical mouse, and an architectural methodology for smart digital sensors," *VLSI Systems and Computations*, H.T. Kung, R.F. Sproull, and G. Steele (eds.), Computer Science Press, Rockville, MD.
- Mead, C. and L. Conway. 1980. *Introduction to VLSI systems*, Addison Wesley, Reading, MA.
- Newman, W. N. and R. F. Sproull. 1979. *Principles of interactive computer graphics*, McGraw-Hill, New York, NY.
- Noll, A. M. 1971. *Man-machine tactile communication*, Ph.D. dissertation, Polytechnic Institute of Brooklyn, Brooklyn, NY.
- Pizer, S. M. and V. L. Wallace. 1983. *To Compute Numerically*, Little, Brown, and Company, Boston, MA.
- Quick, J. H., J. H. Duncan, and J. A. Malcolm, Jr. 1962. *Work-Factor Time Standards*, McGraw-Hill, New York, NY.
- Raab, F. H., E. B. Blood, T. O. Steiner, and H. R. Jones. 1979. "Magnetic Position and Orientation Tracking System," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-15, 709-718.
- Séquin, C. H. and M. F. Tompsett. 1975. *Charge Transfer Devices*, Academic Press, New York, NY.
- Science Accessories 1970. *Graf/Pen Sonic Digitizer*, Science Accessories Corp., Southport, CT.
- Sutherland, I. E. 1965. "The ultimate display," *Proceedings of the IFIP Congress*, Vol. 2, 506-509.
- Sutherland, I. E. 1968. "A head-mounted three dimensional display," *FJCC Conference Proceedings*.
- Tanner, J. E. and C. Mead. January 23, 1984. "A Correlating Optical Motion Detector," *Proceedings, Conference on Advanced Research in VLSI, MIT*, Paul Penfield, Jr. (ed.), Artech House, Dedham, MA, 57-64.
- Ullman, S. 1979. *The interpretation of visual motion*, MIT Press, Cambridge, MA.
- United Detector Technology 1981. *Trade literature on OP-EYE optical position indicator*, United Detector Technology, Inc., Santa Monica, CA.
- Venot, A. August 29-September 2, 1983. "Digital Methods for Change Detection in Medical Images," *VIII Information Processing and Medical Imaging Conference*.
- Vickers, D. L. 1974. *Sorcerer's apprentice: head-mounted display and wand*, Ph.D. dissertation, Department of Computer Science, University of Utah, Salt Lake City, UT.
- Wallmark, J. T. 1957. "A new semiconductor photocell using lateral photo-effect," *Proceedings IRE*, Vol. 45, 474-483.
- Woltring, H. J. 1974. "New possibilities for human motion studies by real-time light spot position measurement," *Biotelemetry*, Vol. 1, 132-146.