An Image Based Focusing System:
An Alternative for Cameras,
a Model of Eye Accomodation

Ted Selker

COINS Technical Report 81-19

(September 1981)

This work done as part of requirements for Honors; Bachelor of Science Degree, Brown University, June 1979.

AN IMAGE-BASED FOCUSING SYSTEM

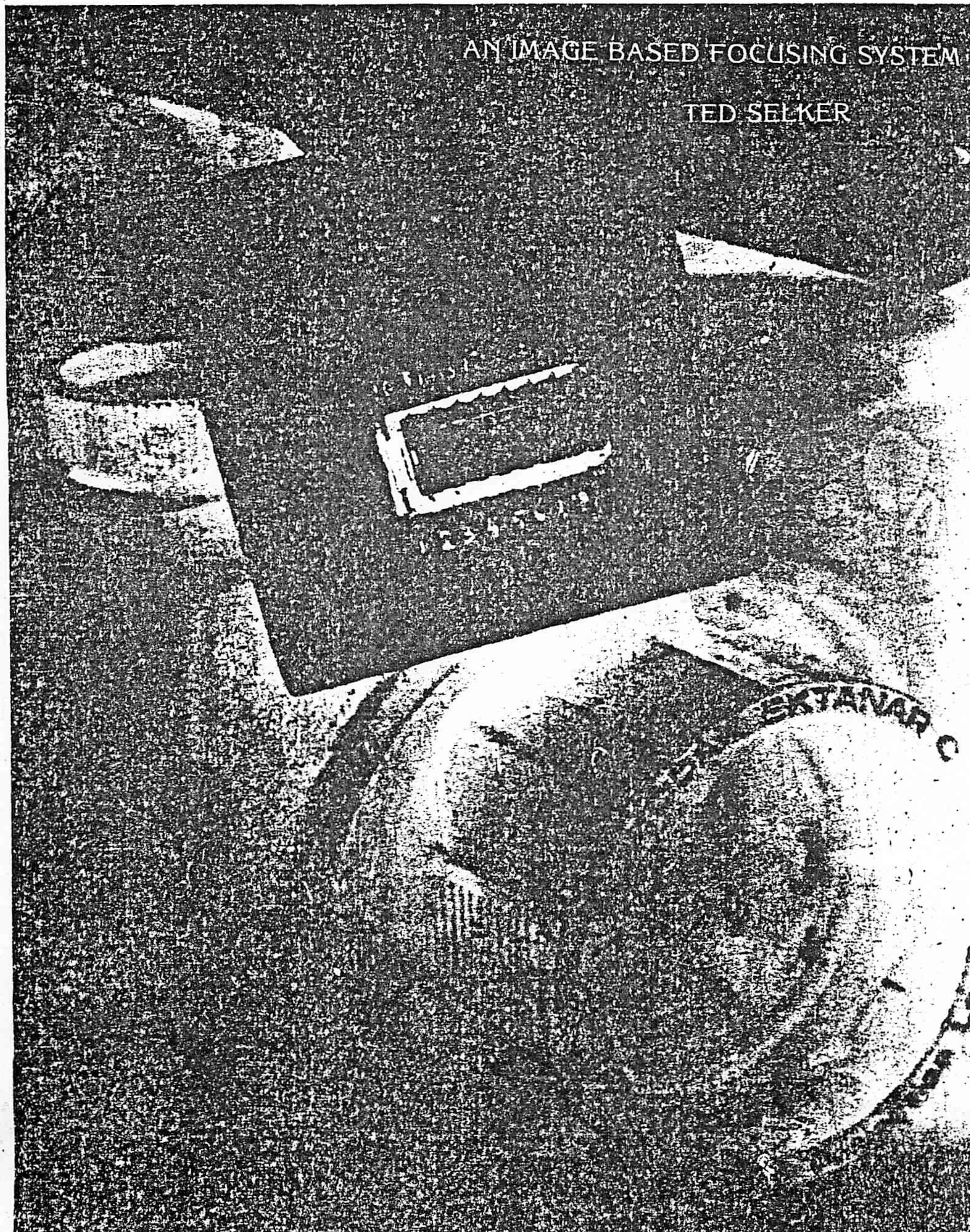
Ted Selker
Brown University
Providence, Rhode Island

Abstract

Hardware simulating a retina was developed to test whether lateral inhibition is an adequate processing basis for the determination of best focus in a dense sensor imaging system on flat fields. The system developed shows that maximizing the sum of adjacent sensor light level differences is an adequate algorithm for focusing on a wide range of visual stimuli.

The hardware was developed as a prototype image focusing and retrieval system. A Reticon light sensitive array (Reticon RL256G) is used as the sensing element. An Intel 8085 based computer with limited memory (512 bytes ROM,512 bytes RAM) is used to process the data. The system automatically focuses on stimuli in object planes. It can also manually focus, input images and take light level and spatial frequency measurements of scenes.

Besides being well suited for studying ocular contrast difference judgments and computational problems of image interpretation, this system is appropriate for video and film applications. On a movie or video camera, the system could meter light, focus and in many cases track specific moving objects. This would give the camera operator increased control of a scene.


AN IMAGE BASED FOCUSING SYSTEM
TED SELKER
EKTANAR

INTRODUCTION

Ocular focus for humans and animals is in most cases an unconscious effort. Only in the case of organic disabilities do people have trouble focusing on stimuli in normal visual experiences. With effort, people can defocus the image falling on their retina. This paper describes a project which explored:

1) What does "best focus" mean?

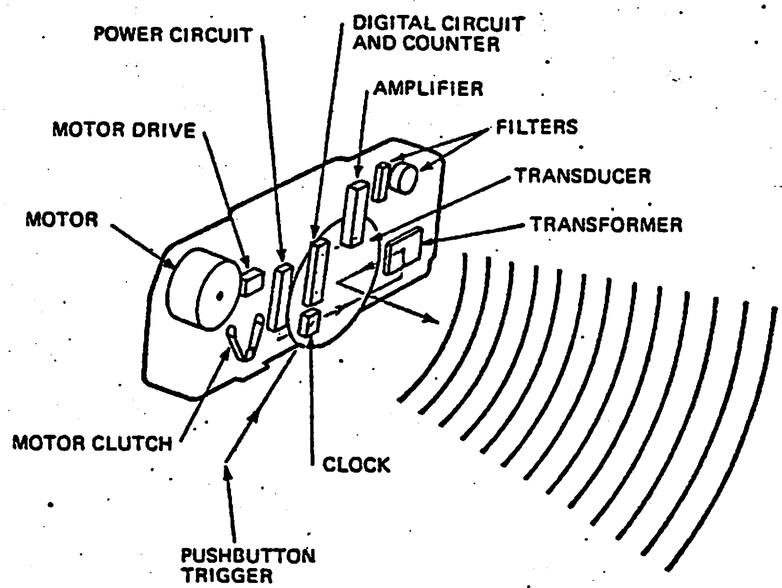
2) What is the simplest processing required to use information from an image in order to optimize an object's focus through a lens?

3) What is the difficulty of constructing a focusing aid designed to save video camera operators effort?

Two automatic focusing systems are now commercially available (Figure 2). Honeywell announced their "Visitronic" automatic rangefinder style focuser in 1978. This system uses 4 matched pairs of light sensors located behind a special rangefinder-style mechanical and optical setup. Comparators signal a solenoid to move a rangefinder mirror and lens to correlate light levels on matched sensors. Polaroid introduced a sonar-focusing measurement system on a new line of cameras in the same year. Their system reflects a focused beam of ultrasonics off of an object and measures its travel time to determine object distance. Neither of these focusers is self-calibrating; neither relies on the

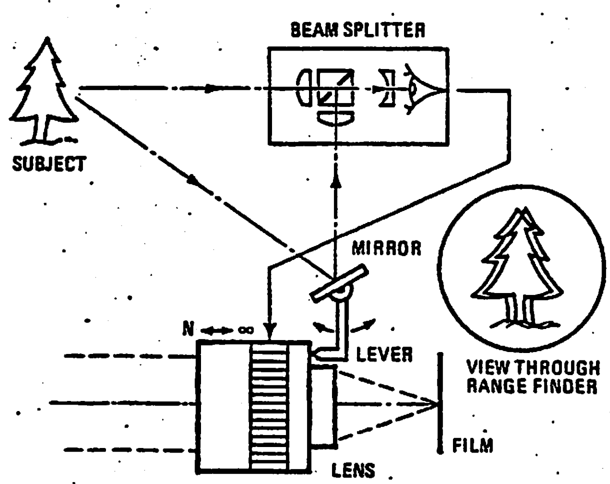
Two Commercially Available Non Image Based Focusing Systems

Figure 2



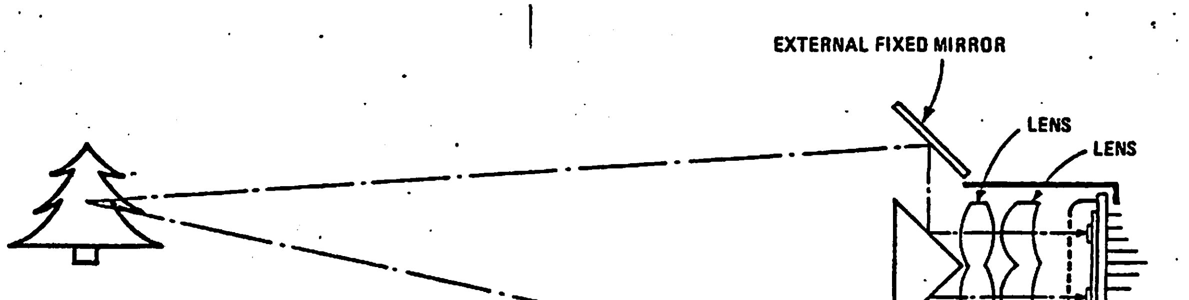
POLAROID

Ranger. Sound waves created by electrostatic transducer helps new camera find range. Motor focuses lens according to time for sound to hit target and return.



HONEYWELL

(a)



(b)

1. In the picture. To focus a conventional range-finder camera, an optical-beam splitter (a) splits the image and a movable mirror linked to the lens is adjusted until the images coincide. The autofocus system (b) uses two mirrors to reflect two images onto two photo-sensor arrays. An IC establishes when the two images are matched and creates a correlation signal to control focusing.

actual image projected.

An image-based focusing system uses the image projected through a lens as its database. To focus using this information one needs to understand focus. Ideal focus exists when the light level (which we will call luminescence) distribution function on the image plane varies from the stimulus (object) plane's luminescence distribution only by coefficents of enlargement and brightness. In this condition with a magnification of 1, object plane points would map directly onto image plane points. Defocus is characterized by points in the object plane mapping onto a distribution area about the place where the focused point would map in the image plane. This area is what we call the blur-circle on the defocused image. The luminescence (L) on the image plane corresponding to a light point of light level (E) on the object plane can be described as

$$L= E \; 1/(2\,PI\,(3@))\,e**<-1/2(r**2)/(3@)**2> .$$

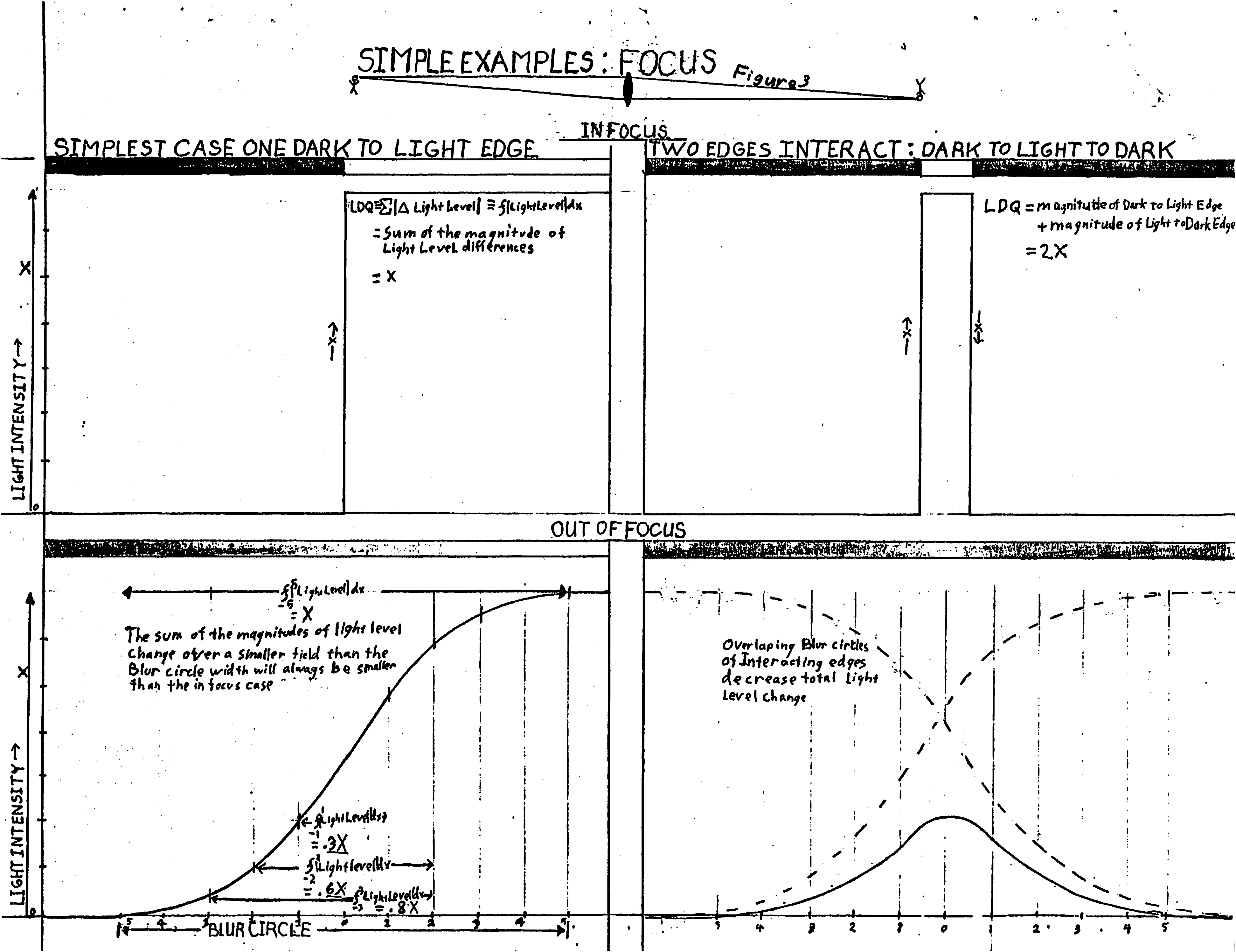
R is the distance away from the center of the area to which an object point is mapped and @ is the distance (r) at which 67% of the light is within r. (Blur Of The Retinal Image, page 14). The larger the blur-circle is, the worse the focus. This blur affects different scenes in different ways.

If an object plane is all one luminescence, it is not possible to determine ideal focus. The redistribution of object plane light points onto image plane blur-circles will only change the source of image light, not the actual image light level. Light from adjacent object plane points will overlap in the image plane. Since these all have the same intensity, no fluctuations in light level will be introduced.

The next simplest focus problem, that of 1 light level change, can be most easily considered in the one-dimensional image plane. Consider an image plane which is a line. If part of it is light and part of it is dark, there will be only 1 light level change in ideal focus; that of the interface (Figure 3A). This change is equal to the difference in brightness of the two parts of the stimulus. If the system is defocused, blur-circles of all points from light side within blur-circle radius of the interface will map their light to the dark side of the interface. There will a gradient of light level change about the interface. Within this radius on either side of the interface, the light level will be changed; the light side darkened as well as the dark side lightened. Contrast (change in luminescence with respect to distance divided by luminescence) is decreased. For points along the line far enough away from the step light level change, light level is unaffected. If the blur-circles from the step light change map off the image plane, the total light level change along the line will be decreased.

A more complicated situation consists of two opposing light level changes along the one-dimensional image plane. A good example of this is a bar of light (Figure 3B). In this case, if the blur-circle caused by defocus is larger than 1/2 the width of the light bar, the maximum light level of the bar in the focus condition will not be attained. In this case all points on the light section spill light onto the dark sections. If the image projection of the bar is within blur-circle radius of either edge of the image field, the maximum darkness of the focused condition will not be attained either. In all of these defocus cases, the sum of the absolute values of light level changes along the image plane is decreased. Even in the cases where defocus does not cause a decrease in absolute light level changes, it does decrease contrast.

We call the sum of the absolute value of light level changes along the image line the Luminescence Difference Sum (LDS). The sum of squares of the differences of light level change across a scene will be decreased with defocus even for simple decrease in contrast. We call this the Luminscence Difference Square Sum (LDSS). The more complicated an object stimulus is, the higher its spatial frequency and the easier it is to see change in the LDS with


SIMPLE EXAMPLES: FOCUS
Figure 3
IN FOCUS
SIMPLEST CASE ONE DARK TO LIGHT EDGE
TWO EDGES INTERACT: DARK TO LIGHT TO DARK
LIGHT INTENSITY →
LDQ ≡ Σ|Δ Light Level| ≡ ∫|Light Level|dx
= Sum of the magnitude of Light Level differences
= X
LDQ = magnitude of Dark to Light Edge + magnitude of Light to Dark Edge
= 2X
OUT OF FOCUS
∫|Light Level|dx = X
The sum of the magnitudes of light level change over a smaller field than the Blur circle width will always be smaller than the in focus case
∫|Light Level|dx = .3X
∫|Light Level|dx = .6X
∫|Light Level|dx = .8X
BLUR CIRCLE
Overlaping Blur circles of Interacting edges decrease total Light Level change

defocus. For most real life focus problems it appears that that maximizing this LDS gives best focus.

For an array of discrete detectors such as a retina, an adequate algorithm for focus simply takes the LDS over the array and maximize it by changing focus. A system was built to test this hypothesis and to explore uses of it.

METHOD

A focusing system was conceived to simulate human methods of focusing in an artificial retina. Investigative purposes included: testing a model we had proposed concerning a possible way in which animal ocular systems focus themselves, creating a general-purpose versatile focusing and imaging system capable of tracking focusing, and analysing scenes. Several configurations for the "retina" were proposed:

1) A tightly packed arrangement of light sensing diodes would allow parallel outputs which could be processed like that of a retina. In this system the outputs of adjacent light sensors would be physically connected to layers of parallel circuitry. This circuitry would first introduce lateral inhibition (the tendency for stimuli presented next to a sense field to decrease its sensitivity) between sensors. This sensing circuit would then have a contrast sensitive output.

2) Improved sensor density could be realized by feeding the sensors with a fiber-optics bundle. with a fiber-optics bundle to increase the resolution of the image. Closely packed fiber ends would map to widely spaced light sensitive diodes.

3) Another sensor was also considered which utilized a serially addressable diode array as a high density central vision area and discrete devices for peripheral area These

areas correspond to areas in the human eye.

4) A simpler system with one densely packed horizontal line of 256 sensors using a Reticon RL256G was a compromise.

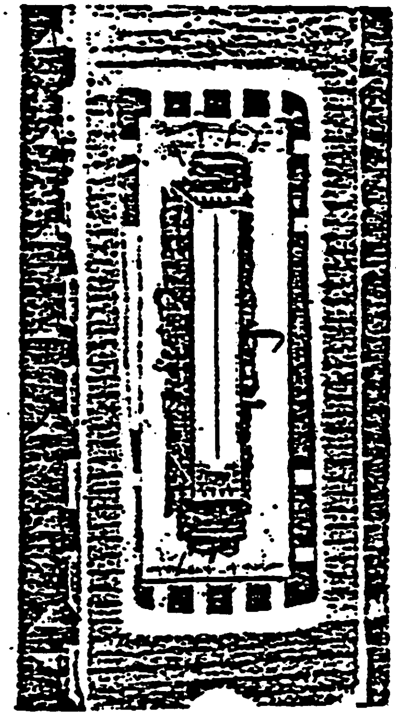
While more physiological, parallel-processing models can only be tested with discrete elements, they can be simulated with a serial device. The computational problems for a two-dimensional input would be impossible to handle with the computers readily available; hardware realizations would be equally impractical. The Reticon serially addressable array was chosen. It has light-sensitive diodes mounted every 15 microns for 0.25 inches. Other horizontal hardware configurations could be simulated using it.

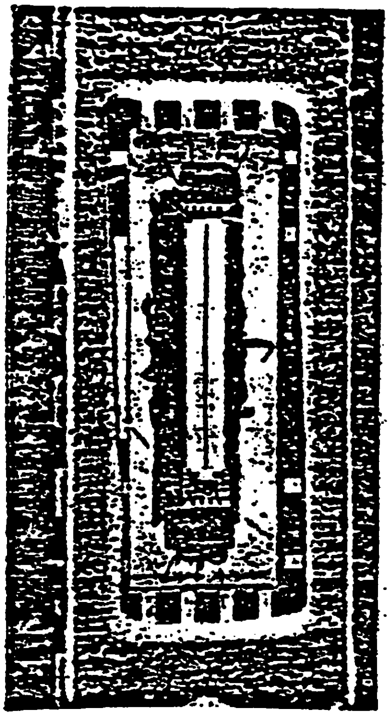
HARDWARE

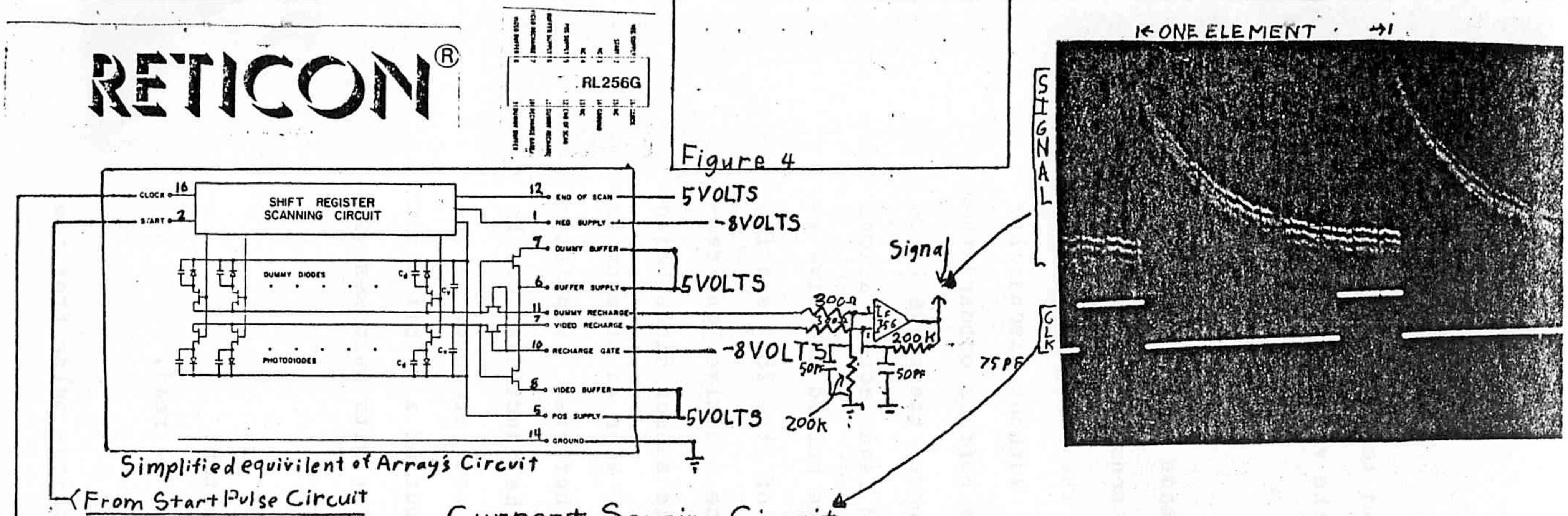
To become familiar with the Reticon array, more than one method of data acquisition was tested. Measuring the voltage level on the sensors is the most stable, versatile output (but the hardest to arrange). Measuring the current discharge of each sensor is easier but less useful. First a circuit to display current discharge of each element on an oscilloscope was set up.

The Reticon chip consists of a shift register whose 256 outputs are connected to Metal Oxide Semiconductor (MOS) switch transistors. Light sensitive diodes deposit charge on capacitors. When the MOS transistors are off, the sensors are isolated from the rest of the circuit. When an MOS switch transistor is turned on, charge deposited on the associated junction capacitor is connected to the outputs of the chip. At the same time, these are connected to identical outputs for reference purposes. The chip addresses one of these sensors at a time. This allows them to charge for 255 cycles before being read. Another identical set of sensors not exposed to light is selected in parallel with the light sensitive array. The outputs from this array are subtracted from the signal to eliminate timing noise. The output data line of the chip goes through 2 MOS transistors. The data line is connected to the gate of a buffer transistor for high impedance voltage measurements. The data line is also connected to the source of a recharge gate to remove the light-deposited charge from the sensor junction capacitors each time they are read. Since the array only addresses a specific sensor once every 256 cycles, the sensitivity is inversely proportional to the speed of the clock and directly proportional to the time the sensors have to charge between readings. If this array scan time is over 1/30th of a second, however, noise rises above 1/10 of signal level and the light level data will have significant offset error. Each of the 2 methods of extracting data from the array uses its circuitry differently.

For the current sensing method of data acquisition, chip drive requirements include: a start scan pulse, a clock pulse, and a difference amplifier for output (Figure 4A). A Tektronix signal generator was used for the clock pulse generator. Two 4 bit up-down counters (74192's) were used to divide the generator's output by 256 to serve as a start pulse. An LF256 instrumentation amplifier was used as a buffer and difference amplifier between the dummy diode and the data diode output current. These currents were sensed across load resistors. The buffer transistor supplies were wired off and the recharge gates were biased on. The load resistor was connected directly to the drains of the recharge buffers. Using this system the sensitivity of the array was shown to be within acceptable tolerance of the manufacturer's specifications. It is difficult to obtain

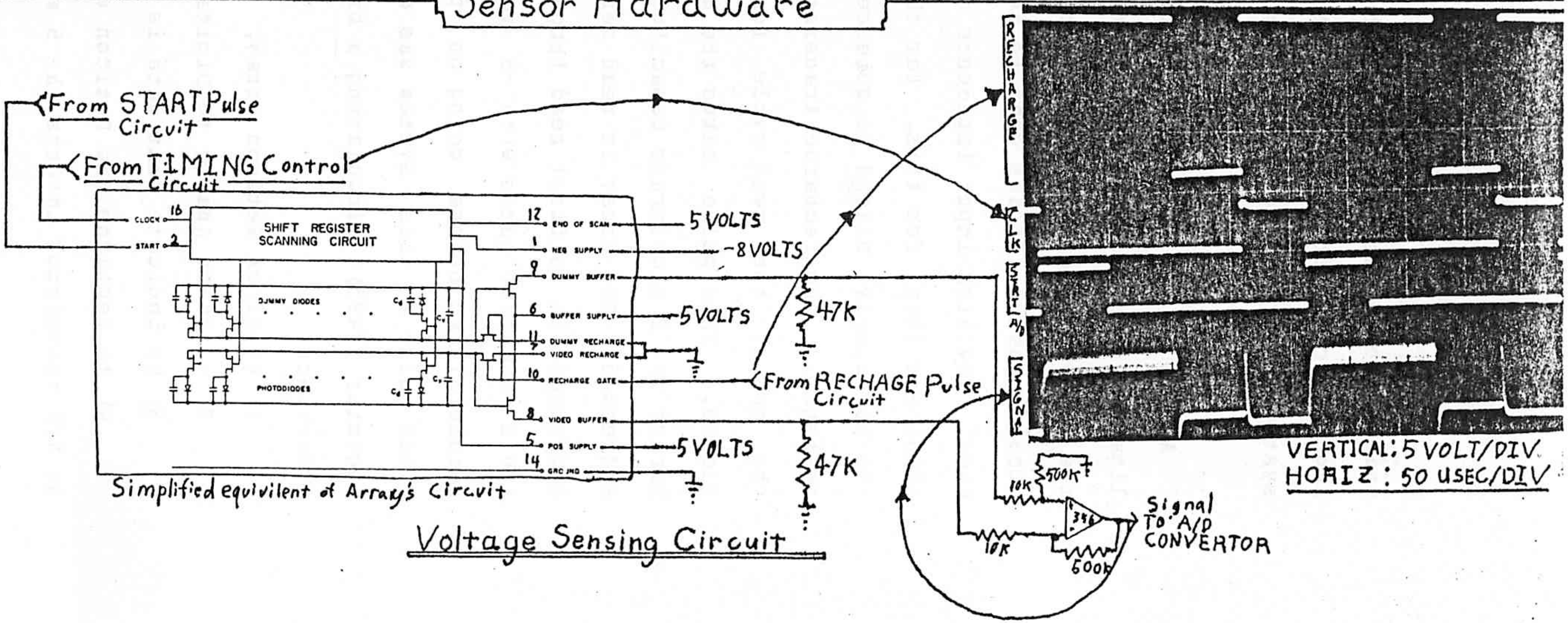






Current Sensing Circuit

Sensor Hardware



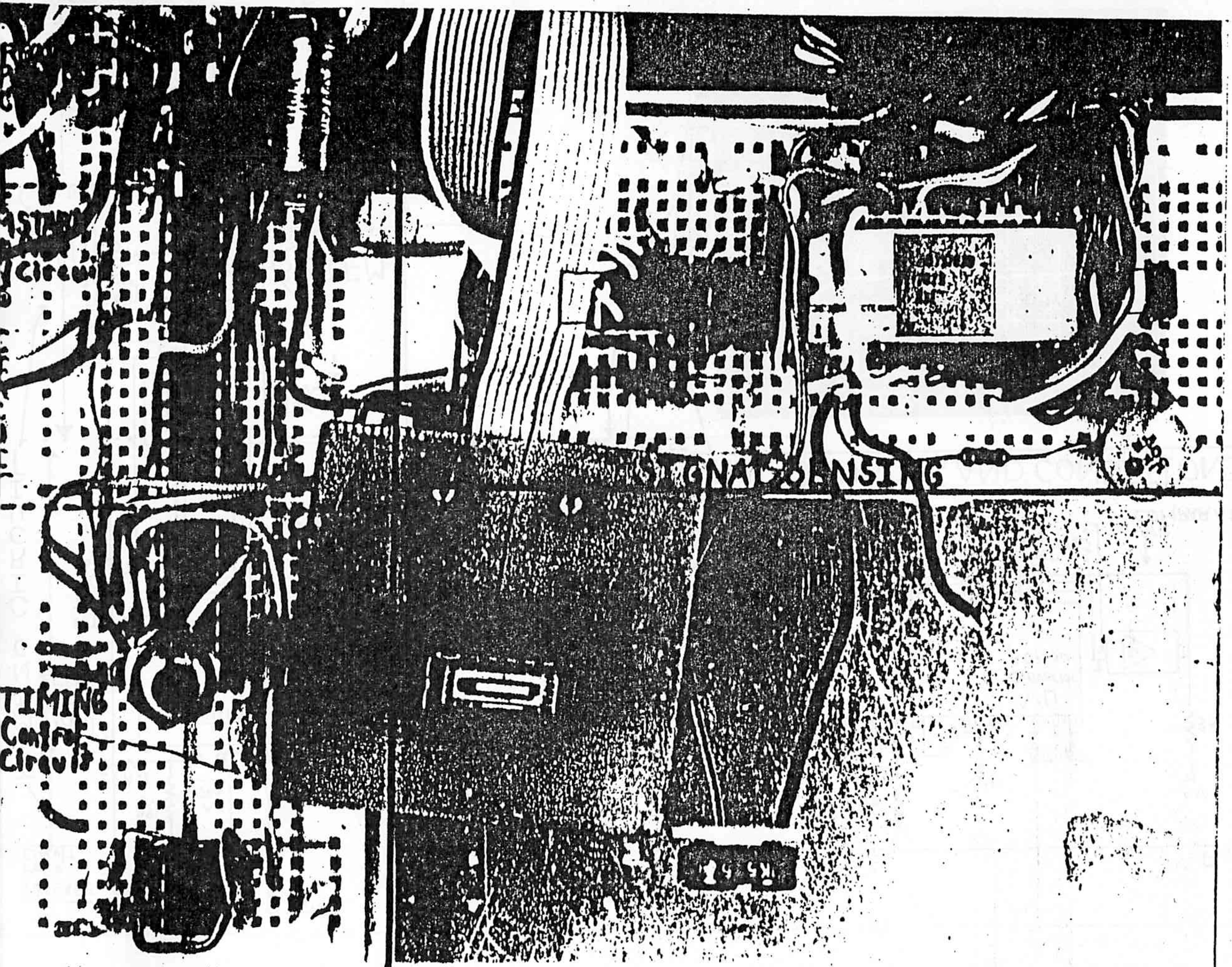
Voltage Sensing Circuit

stable samples using the current sensing technique of reading the array, it was hoped that using the voltage sensitive system would improve performance.

A circuit was designed to put a voltage level on a data line to correspond to illumination of sensing elements of the RL256G (Figure 4B and Figure 5). The charge potential across sensing diodes can be measured without discharging them. The high input impedence dataline buffers onboard the LF256 are ideal for this. For this purpose, these MOS buffers are simply biased to quiescence and read across a load resistor. The recharge transistors are pulsed negative at the end of the read cycle for each of the 256 sensing diodes. This pulse makes the data-line negative and recharges the 2 picofarads capacitor on the sensor diode being addressed. The array is read serially. When an element is addressed, it is first read through the buffers. By pulsing the 2 recharge gates with -8 volts, the junctions can be reinitialized before going on to the next element. The clock pulse in this system was one output of a 4 bit shift register (7495) circulating a bit. The bits purposes (in order) are:

1) to clock Reticon array,
2) to start Analog to Digital Converter,
3) to indicate that data is ready to be read,
4) to recharge the Reticon elements.

An NPN transistor inverts the 5 volt recharge pulse from the


SIGNAL SENSING
TIMING
Control
Circuit

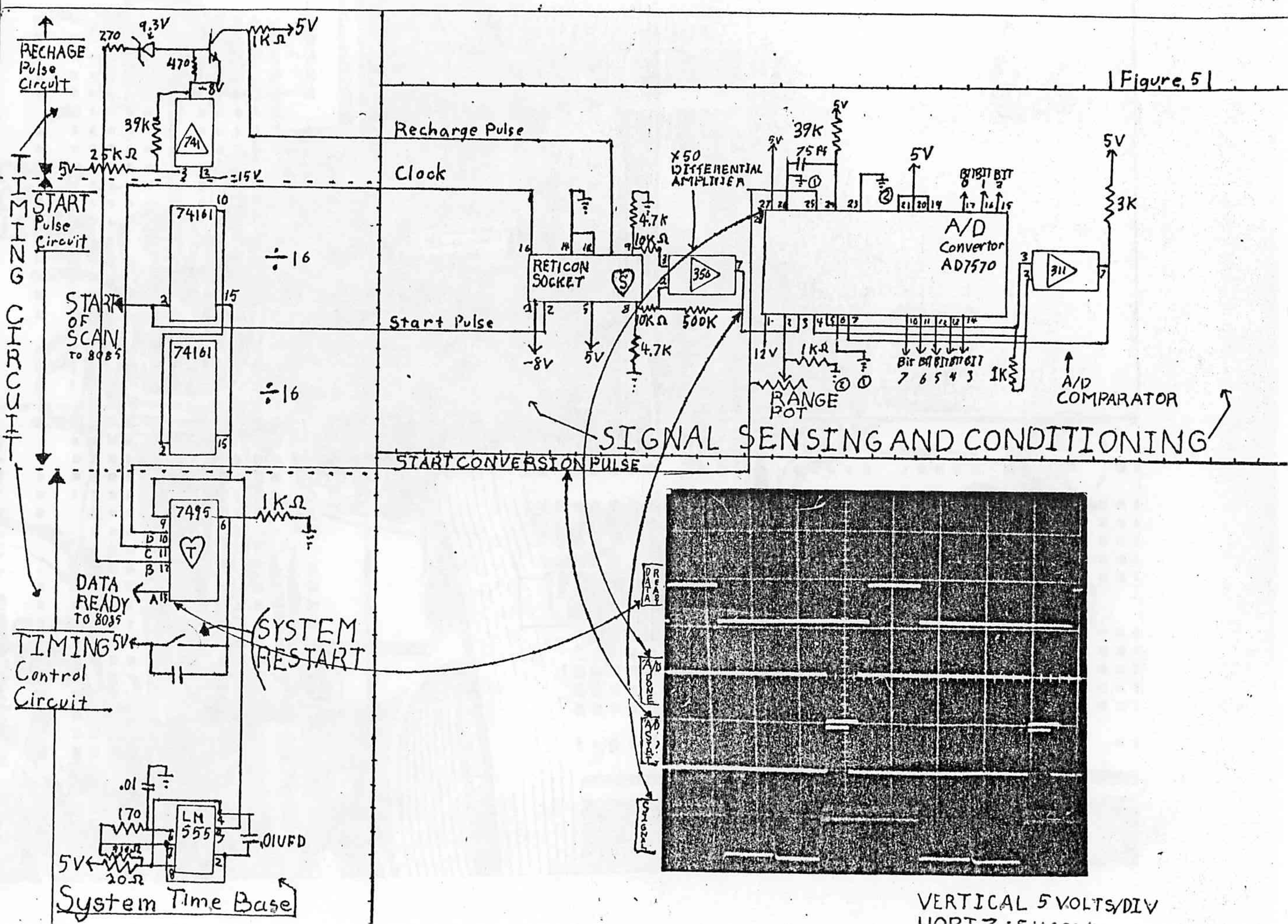


Figure 5
TIMING CIRCUIT
RECHAGE Pulse Circuit
START Pulse Circuit
TIMING Control Circuit
System Time Base
SIGNAL SENSING AND CONDITIONING
Recharge Pulse
Clock
Start Pulse
START CONVERSION PULSE
START OF SCAN TO 8085
DATA READY TO 8085
SYSTEM RESTART
74161
÷16
7495
LM 555
741
RETICON SOCKET
X50 DIFFERENTIAL AMPLIFIER
356
A/D Convertor AD7570
311
A/D COMPARATOR
RANGE POT
Bit 7 6 5 4 3
BIT 0 1 2
.01UFD
DATA READY
A/D DONE
SIGNAL
VERTICAL 5 VOLTS/DIV

shift register to make a
-8 volt pulse to activate the recharge transistor.

The signal is sensed through the buffer transistor by applying 5 volts to the buffer's source and sensing across a grounded resistor. Optimal signal was obtained when this resistor was 4700 ohms. Even when maximum swing was assured by correctly biasing the Field Effect Transistors, the maximum voltage signal output was only .15 volts. The literature had claimes that one could expect up to 3 volts to accumulate on the diodes. The current sensing setup had indicated that the sensor followed the literature specifications. The possiblities of a damaged array, damaged buffers or damaged array elements were all tested.

The array elements' sensitivity was determined empirically, using a laser source and then more completely using a tungsten light (2870 degree Kelvin) (Figure 6). Sensitivity and linearity were both confirmed to be well within acceptable tolerance of the manufacturers specifications. A movable slit was arranged to demonstrate that the shift register was working. The buffer transistors were characterized by connecting a variable voltage to the activated recharge transistors and sensing the output voltage. In this way it was determined that these transistors both worked; they had acceptable leakage and gain characteristics. The recharge transistors were determined to work by measuring current

RETICON ARRAY DIODE LINEARITY TEST

Figure 6

Since light intensity is inversly proportional (L) to the square of distance from (r²) (B) current charge on a light sensitive diode (I) should be proportional to light intensity:

$$L \cdot \frac{1}{r^2} = K \cdot I$$

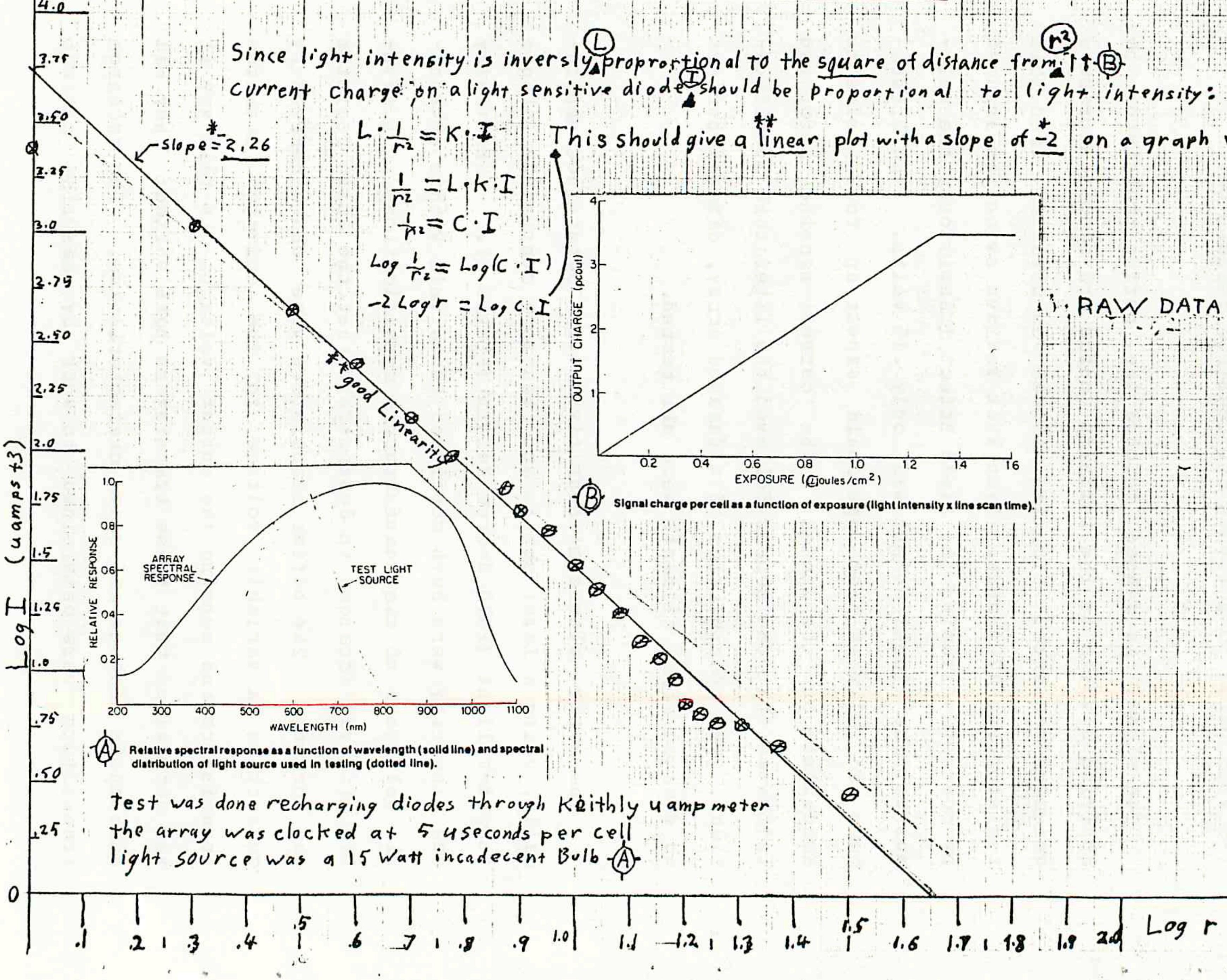
$$\frac{1}{r^2} = L \cdot K \cdot I$$

$$\frac{1}{r^2} = C \cdot I$$

$$\log \frac{1}{r^2} = \log (C \cdot I)$$

$$-2 \log r = \log C \cdot I$$

This should give a linear** plot with a slope of -2* on a graph where X = Log r, Y = Log I



(A) Relative spectral response as a function of wavelength (solid line) and spectral distribution of light source used in testing (dotted line).

(B) Signal charge per cell as a function of exposure (light intensity x line scan time).

test was done recharging diodes through Keithly u amp meter
the array was clocked at 5 u seconds per cell
light source was a 15 Watt incadecent Bulb (A)

RAW DATA

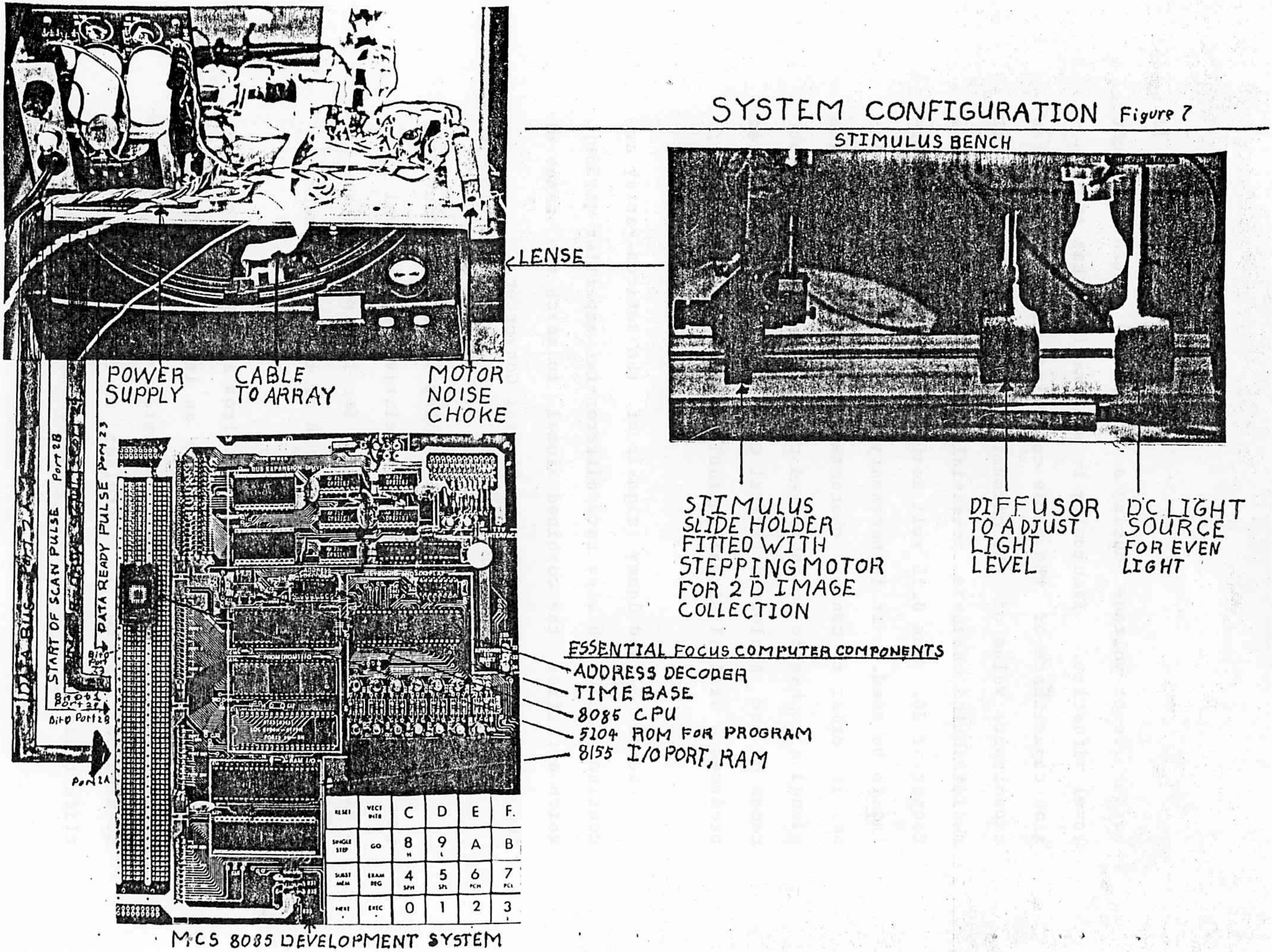
Distance from Bulb	light intensity	Video recharge current
cm	u Joules	u amps
10	175	2.42
20	43	1.22
30	19.3	.45
40	10.9	.225
50	7	.141
60	4.8	.094
70	3.5	.066
80	2.7	.051
90		.040
100	1.75	.03
110	[illegible]	.024
120	1.2	.018
130	[illegible]	.015
140	.89	.012
150		.009
160		.008
170	.605	.007
180		.006
190		
200	.44	
210		

versus reset voltage during a completely saturated light level situation. Eventually it was realized that the junction capacitance of the diode capacitor was about 1/20 the capacitance value of the circuit reading its charge. The redistributed charge's potential decreased the signal by a factor of 20. The 0.15 volt Reticon signal needs conditioning to be used. It is necessary to subtract the dummy signal in order to cancel characteristic timing noise. The signal also has to be amplified to match the 0-10 volt input range of the Analog to Digital Converter. This converter produces a digital signal suitable for input to a computer.

The video and dummy signals of the Reticon array are combined in a high slew rate differential amplifier (LF356), which amplifies the combined signal to match the range of the next stage, the Analog to Digital Converter.

An Analog Devices AD7570J Analog to Digital Converter is used to convert the elements' relative output level to an 8 bit number. The chip was chosen for its speed and was run off its own clock with a cycle time of twenty microseconds.

The optics and focusing motor from a 750H Kodak carousel projector was chosen to project an image onto the array (Figure 7 shows the system configuration). The array was mounted on a piece of fiberglass in the shape of a 35mm slide holder. In this way light projecting into the front



SYSTEM CONFIGURATION Figure 7

of the projector could be accumulated from the position of a normal slide. The projector has a 102 mm lens on it. The 4.02 inch focal length determines the angle of visual field which will project onto the 0.25 inch photosensitive array when the lens is focused at infinity. This gives a field with an

arc =arctangent 0.25/4.02 =3.56 degrees.

To use a 35 mm slide as the object plane it had to be positioned a distance

= sin(3.56)35mm =1.5 feet away from lens.

The focus motor requires +12 and -12 volts to move it forward and backward (see Figure 6). A 2 bit Transistor Transistor Logic (TTL) compatible driver was needed to use the 8085 TTL compatible output ports to drive the focus motor either direction or keep it still.

Using both inputs of a 741 in a difference amplifier configuration with an amplification of 3, both inputs low or both inputs high drove the motor with 0 volts. With the positive input of the amplifier high and the negative input low, the motor was driven with 12 volts. With the negative input of the amplifier high and the positive low, the motor was driven with -12 volts. Since the 741 cannot handle the full 120 milliamperes load of the motor, a monolithic driver amplifier was used to drive the motor (Figure 8A). This expensive device was replaced with VFET transistors (Figure 8B). These were found to be temperamental. The application

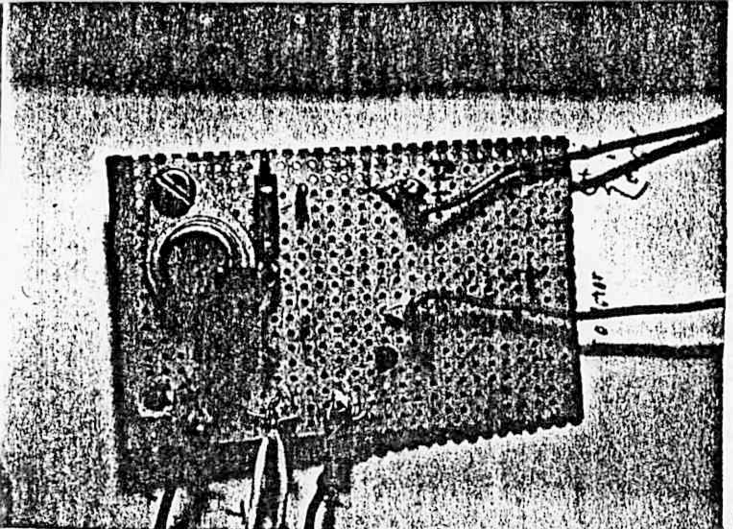


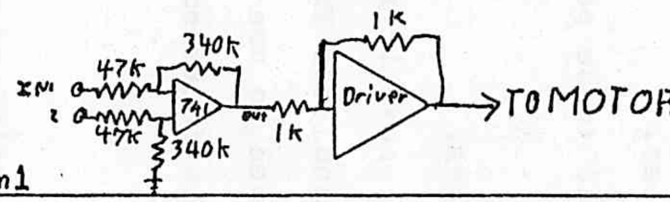
Figure 8

3 Bidirectional Motor Driver DESIGNS

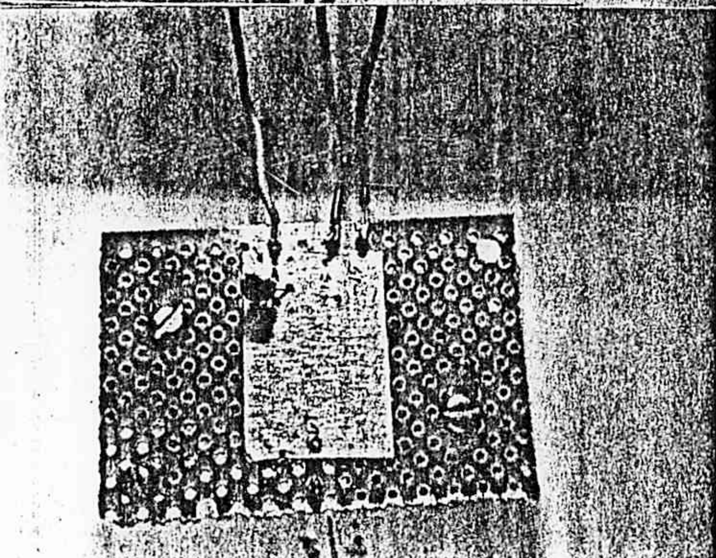
INPUT: Zero -or- One TTL, 2 Bits

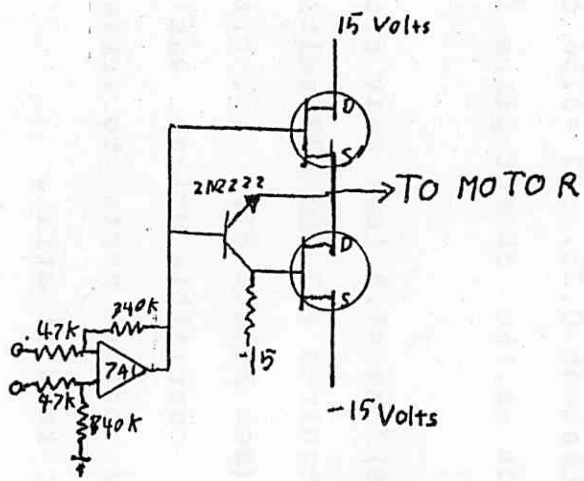
OUTPUT: 12 VOLTS, -12 VOLTS or 0

120 Milliamps



Design 1

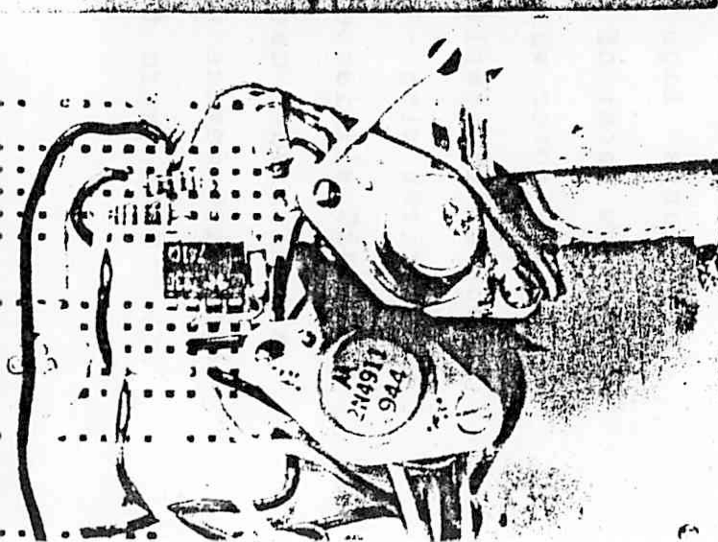


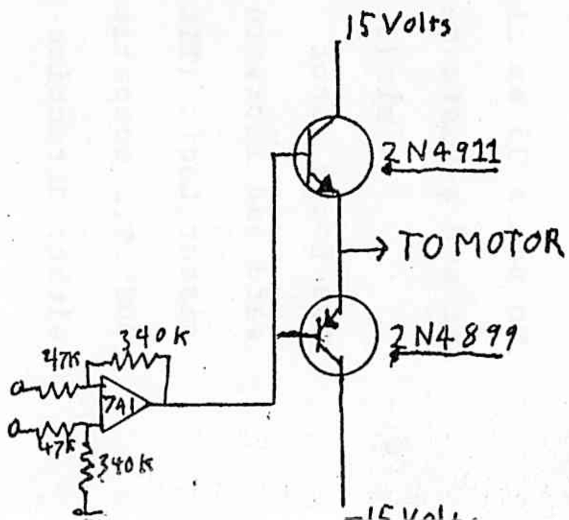


2

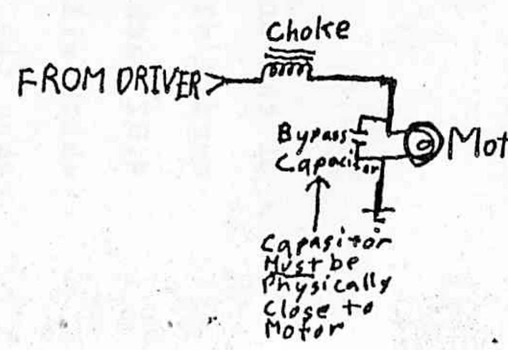
741 Truth TABLE

IN1	IN2	OUT
0	0	0
0	1	-15V
1	1	0
1	0	+15V





Projector Motor and Motor Noise Elimination Circutry



of a pair of power transistors; an NPN and a PNP was the cheapest and most reliable solution (Figure 8 C).

A problem arised when the Reticon array's circuit and the motor circuit were run simultaneously. The motor generated pulses on the power line which affected the timing circuits. A capacitor was put in parallel with the motor to bypass noise. Only when such a capacitor was placed directly on the motor was it effective. In other cases the lead wire in series with the capacitor acts as a resistor which gives the filter a time constant (RC). High frequency noise is unattenuated. A 0.2 microfarads (UF) bypass capacitor directly across the motor and an inductor in series with it together were effective in eliminating motor noise on the power supplies.

If a suitable computer is used, a program can run fast enough to leave the sensing system described above free running. The LDS could be calculated for the serial outputs of the elements of a light sensitive array with no backtracking. The National SCMP and the INTEL 8085 were 2 available computers. The speed of the 8085 was needed. 5204 Programmable Read Only Memory (PROM) chips were used as program space. The 5204 PROM is not designed to run fast enough to be used with an 8085. It was found, however, that 5204s could be hand picked which would run fast enough to be acceptable. The computer which was used, an MCS 8085 development system, had 512 bytes of RAM, 6 ports and 512 bytes of program memory in the 5204 which was wired in as program space (Figure 7).

SOFTWARE

Our eyes contain "horizontal" and "amacrine" cells which exhibit lateral inhibition. These cells interact with rod and cone outputs to give an increased output on retinal ganglion cells when adjacent sensors have different light intensities illuminating them. The highest contrast image, the most focused in these cases, gives edge sensitive neurons the greatest output. When signals of this kind are most active in the fovea, the object focused on is in focus. In the periphery, areas with high activity of contrast sensitive "cells" indicate visual areas with a high amount of light level change or changes. A purpose in developing a serial focusing system was to simulate the lateral inhibition with successive difference measurements. Summing the absolute values of these differences gives the LDS. A program was written to maximize the sum of the absolute value of light level changes over the image plane (LDS) by focusing a lens in and out.

A package of programs was designed to develop, test and run the system. The routines of this program library were designed modularly to interdepend (see Appendix A for copies of the documented assembler language code). The copy used to run the programs resides as machine code in 5204 (a 512 byte PROM). The following is a list of the routines in this package, their purpose, design, and alternatives.

P R O G R A M S

UTILITIES

This is a group of support programs. It is used extensively by the other programs in the system.

INIT

This routine is used to initialize the computer to be compatible with the hardware. It must be run before any other program in the package. To use any other routine in the library:

1) Set the 8085 stackpointer to 20C8.
2) Call INIT.
3) Call the routine to preform the desired function.

This routine initializes the motor direction constant to zero (stopped). It initializes the interrupt masks to off. It initializes ports:

21 is an input port. Its first bit is used to sense end of focus range.

22 is used as an output for transfering data to another device.

23 is an input port for the dataready input pulse.

29 is an output port. Its first 2 bits are used to run the motor.

2a is an input used to get the digitalized light level data.

2b is an input port whose low order bit is used to get the start of array line reading pulse.

Design alternatives included the redundant practice of including this routine at the top of each high level routine (not enough space existed) or simply initializing the system in a user inputed routine.

GTSTRT

This routine is used by all routines which interface with the Reticon. It loops until the hardware gives a start of line pulse.

Alternate design would include interrupt driven software or replacing all of the timing circuitry in Figure 5 with software. It was unclear what computer was going to be made available for this project. To make the system most versatile; we made the interface machine independent.

GTRDY

This routine is used by all routines which use the Reticon array. It loops until an element is available for reading.

REGINIT

This routine zeros registers a, b, c, d and e.

MOTOR

This routine exclusive-ors the contents of the motor direction constant with 03. If the constant is 1 indicating forward focus, it changes it to 2 indicating reverse focus. If the constant is 2, it changes it to 1. When direction is changed the last Lateral Difference Sum (the focus constant) is set to zero. The motor control lines are found on bits 0 and 1 of port 29.

MINMAX

This routine is used to find the minimum and maximum light level elements in an array.

END

This routine is used to reverse the lenses direction at the end of its travel. When called, it polls the first bit of port 21. If this bit is high, the routine reverses the direction of the motor.

DEMONSTRATION PROGRAMS

DELAY

This routine waits for an amount of time controllable by registers D and E. It waits a time proportional to (FF-register E)(FF-register D) before returning. It also displays register E to the user. The routine is used to introduce hysteresis in focus programs. Register pair DE is the focus value, this routine will wait for a significant focus change to return, when focus is bad and will cause new focus state evaluation often when focus is close to being the best focus.

READEM

This routine reads light-level values from a user defined Reticon element with addresses between 0 and F. It writes out the elements' value and waits until the user either pushes the execute button or another address number to reread light level.

MOTDEM

This routine can be used to manually focus the lens or to test out the motor system. It simply reads a number from the users' input keyboard and outputs this value to the motor port continuously. It also writes the value to the Light Emiting Diode readout.

INPUT AND OUTPUT

STBUFF

This routine stores a table of light levels from the Reticon array in locations 2800 to 28FF. It waits for the start of line pulse, then begins to read data points. It reads the whole array.

HISTO

This routine puts out the numbers 00 to FF on port 21 the addresses of the data from locations 2800 to 28FF which it sends to port 22. The routine does this in an infinite loop so that if displayed as x and y cn a scope the result is a histogram.

FOCUSERS

SIMPLE FOCUS

This program focuses on a scene using MOTOR to change the direction of the focus motor when the program senses decreasing focus. It uses INIT to set up its constants and initialize hardware. Using preset constants for beginning of window and end of window the program gets a start of line pulse, counts ready pulses until the beginning of the window, and reads the rest of the line up to line length as data. The program uses the simple LDS algorithm to focus. It takes the absolute value of the difference in light level between successive elements and adds this to a 2 byte sum in registers D and E. This sum for each line read from the Reticon is compared to its predecessor unless the program has just changed focus direction.

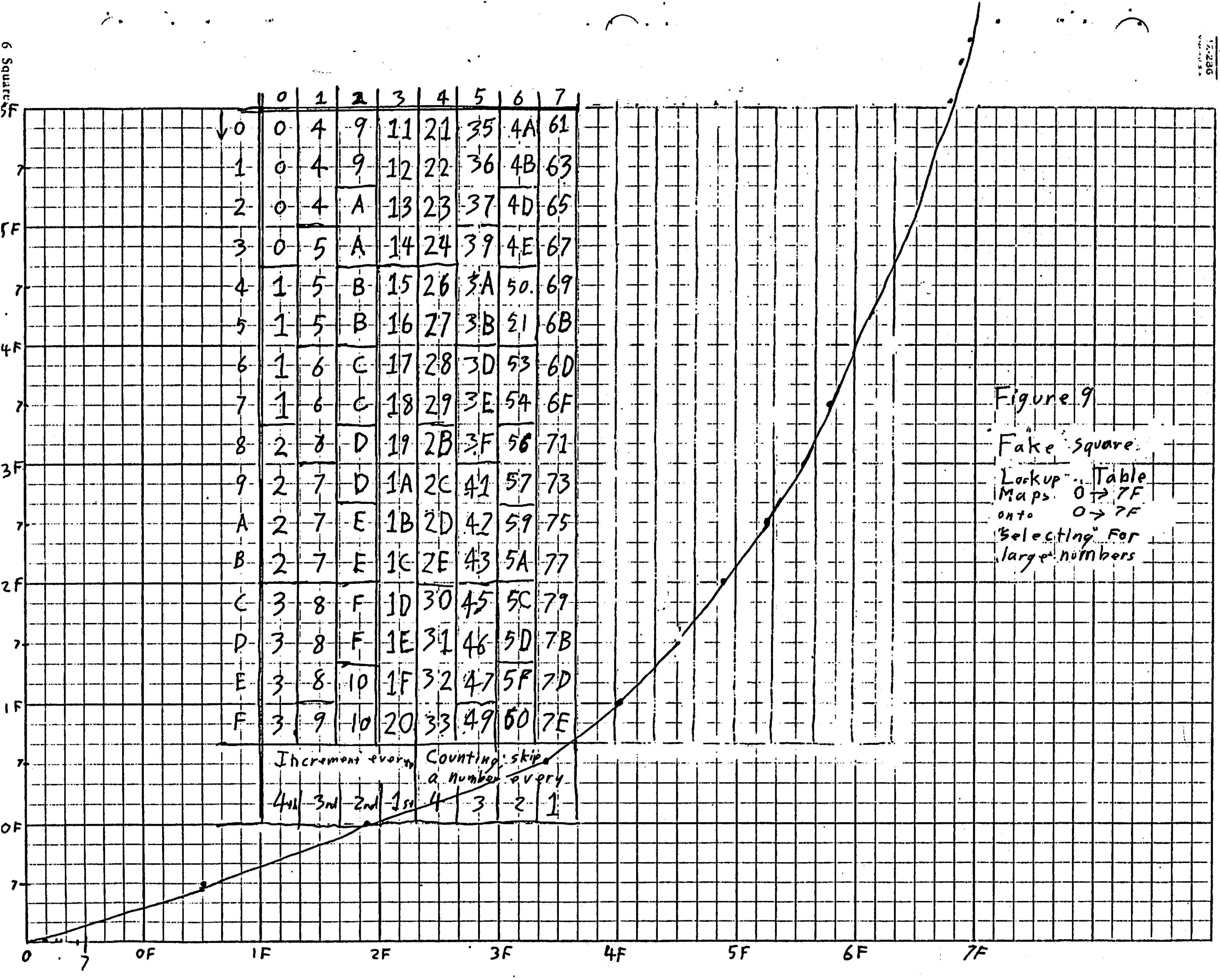
At the end of each iteration, DELAY is called, a user defined program which delays change in focus direction inverse proportionally to the size of the focus metric LDS's magnitude.

FOCUS SQUARES

This program is used as an alternative to SIMPLE FOCUS. It is different in that it takes the square of the differences of light level to improve signal to noise ratio of the system: the LDSS. It uses a lookup table to take the squares. This lookup table is stored in user alterable memory (2800-28FF) (Figure 9). In this way other non-linear transformations can be tested easily.

AUTO FOCUS

This program varies from the other focusers in that it can average data over several elements and can change its focus window width to improve a scene's focusability. It will select a window including the minimum and maximum light intensities along the array. The program selects a resolution and window to focus through, using information from MINMAX about the minimum and maximum light levels seen in the scene. It makes this window overlap the minimum and maximum light level locations in a scene and 1/2 the distance between them on each end if possible.



	0	1	2	3	4	5	6	7
0	0	4	9	11	21	35	4A	61
1	0	4	9	12	22	36	4B	63
2	0	4	A	13	23	37	4D	65
3	0	5	A	14	24	39	4E	67
4	1	5	B	15	26	3A	50	69
5	1	5	B	16	27	3B	51	6B
6	1	6	C	17	28	3D	53	6D
7	1	6	C	18	29	3E	54	6F
8	2	7	D	19	2B	3F	56	71
9	2	7	D	1A	2C	41	57	73
A	2	7	E	1B	2D	42	59	75
B	2	7	E	1C	2E	43	5A	77
C	3	8	F	1D	30	45	5C	79
D	3	8	F	1E	31	46	5D	7B
E	3	8	10	1F	32	47	5F	7D
F	3	9	10	20	33	49	60	7E
	Increment every				Counting: skip a number every			
	4th	3rd	2nd	1st	4	3	2	1

RESULTS

After the system was complete, tests for its abilities were run to evaluate and improve it. LDS numbers were taken for stimuli of different spatial frequencies, including one single edge, 2 opposing edges, people, text and simple spatial frequencies up to 12 lines/cm. The focuser's step response (response for focusing stimuli presented initially out of focus) and tracking response (maximum speed for keeping moving stimuli in focus) were also evaluated.

The mount of the lens projecting the image onto the Reticon array was a problem. When the lens focus direction is changed; the track on which the lens moves alows the lens alignment to change. The lens has a gear rack on one side of it. When the motor pinion differentially loads the rack on the side of the lens, the play in the track is noticed as a rotation of the lens about a line perpendicular to its center line. The magnitude rotation is 1/8 degree (=0.125 degrees/3.56 degrees in field=1/12 of visual field). The alignment of the lens is held relatively constant if the motor direction is not changed. The motor reversing routine was altered to inhibit motor reversal immediately following a motor direction change. This insures that both LDS'S or LDSS'S being compared have been taken since the last motor change. This improved the stability of the focus considerably.

The speed of lens movement in focusing was measured to be about 1.5 mm/second. Its total range has been measured to be about 3 cm. This causes a limitation on the speed with which the system could focus and track a moving stimulus. The focus motor also has an onset latency. This is due to it being a direct current motor working against the inertia of the lens. It was found that focus change measurements had to be delayed just to get valid readings from 2 different focus positions. Using a delay proportional to (FF-low order byte)(FF-high order focus byte) sped focus. It allowed the system to focus large distances without interruption, when far away from best focus. As best focus was approached, the distance between focus reevaluation was decreased.

Three simple focus programs for the system were compared: OLD, NEW and SQUARE. OLD focuses making an Luminescence Differences Sum (LDS) which is composed of the sum of the difference of adjacent sensors. Its field of differences are non-overlapping. NEW, on the other hand computes the difference of all adjacent sensors by overlapping the differences. SQUARE allows the user to apply a lookup function located in the table starting at memory address 2800 to the differences in the LDSS. Because of time and memory constraints computation was confined to a 7 bit word. A function was placed in the SQUARE lookup-table which increases the increment between numbers mapped onto pseudo quadratically (Figure 9). A 3.5 bit lookup table of squares was also tried. The lack of resolution more than outweighed the utility of the algorithm.

Analysis of the outputs of the imaging array on an oscilloscope was useful in obtaining information about imaging systems' limitations. Stimuli were first observed which had only high spatial frequency light level changes. Although when in focus they have large light level changes, when out of focus they may have no significant luminescence changes. From Figure 10; pictures 1, 2, 3, 5 and 8 are all examples of stimulus in this category. Picture 2 is the best example; when it is defocused 10 centimeters in the object plane, the image picked up by the imaging array appears to have imperceptible luminescence change (the change which makes focus possible). The other stimuli mentioned have the same problem. High spatial frequency stimuli have easily distinguishable luminescence change peaks when close to focus. It is not difficult to tell when they are in focus. It tends to be difficult to obtain best focus information for stimuli with low frequency luminescence changes. Their light level peaks are small and LDS comparisons are hard to make. Stimuli with mixed frequency light level changes have the sharp in-focus peaks and do not totally lose their contrast as quickly with defocus. From Figure 10 pictures 4, 6, 7 and 9 are all examples of stimuli in the mixed spatial frequency category. Picture 7, the best example of distrib-

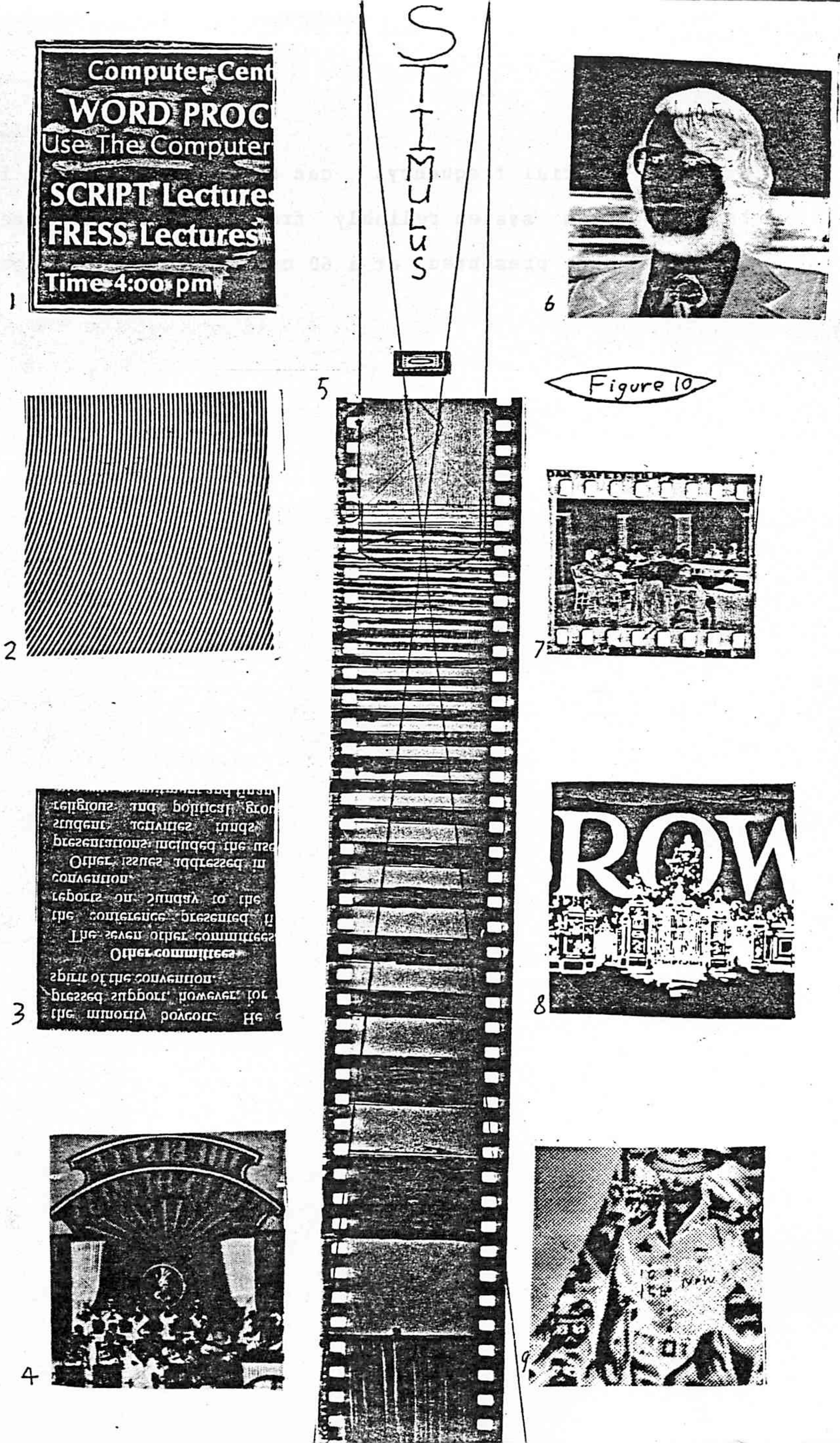

STIMULUS
1
Computer Cent
WORD PROC
Use The Computer
SCRIPT Lectures
FRESS Lectures
Time 4:00 pm
2
3
4
5
6
7
8
ROW
9

Figure 10

ution of spatial frequency, can be focused by the image based focusing system reliably from up to 30 centimeters defocused when presented at a 60 cm standard distance from the lens.

SYSTEM PERFORMANCE

All focus programs were shown to be able to find best focus on all of the stimulus material in Figure 10. Several system limitations were noted, and each of the programs performed better under different situations. High frequency stimulus was focused fastest using the NEW overlapping algorithm. The SQUARE algorithm only worked well with the function in Figure 9 as its "squared table" when the light level was high. This is because the function differentiates data best only for LDS focus numbers bigger than 3F. The non-overlapping algorithm (in OLD) works well for mixed stimuli. With this algorithm more space is left between numbers which are subtracted to make the LDS focus quality number.

Step response for stimulus (the amount of time required to focus a stimulus when initially presented in a defocused state) varied. When the DELAY program was used, fewer wrong decisions were made. If the stimulus was shown to the system within stable focus range (30 cm range at a standard 60 cm object distance for picture 6, 10 cm range at standard distance for picture 2) the focus time would be no more than double the time it took the motor to move to that position at full speed (6 seconds for picture 6). Wrong motor change decisions may be made by the focus algorithm before the system drives itself to within stable focus range. With enough time, focus will be achieved (provided a good range of spatial frequencies is represented in the stimulus).

Tracking response is best for high frequency visual stimuli. This is because the change in LDS is greatest over focus change close to focus for visual stimulus with high spatial frequency. The focuser can track picture 2, for example, at faster than half the speed of the motor's lens position changing ability (1.5 mm/second). All stimuli in Figure 10 can be tracked close to 1/2 the speed of the motor's lens position changing ability.

DISCUSSICN

The results of this autofccus project are extremely encouraging. The entire cost of this prototype Image Based Focusing system is well under $500, includingall optics and the sensor (which are needed for a camera anyway). The system can focus on a wide range of contrived and realistic stimuli. It can gauge light level, spatial frequency and find extremes of light intensity in its field. The system also has user-oriented software to manually focus, display individual image array elements in real time and to read in whole images. Additional software could utilize pattern recognition knowledge to find edges, which could rotate the camera for automatic tracking of moving objects.

The performance characteristics of the prototype described here are inadequate fcr commercial use. With the use of a faster focusing lens moving motor, the system would react proportionally faster (up to one focus change each thirtieth of one second (the time it takes to read the array)). With a better suited ccmputer system and the algorithm could be more easily improved in resolution, could implement an LDSS square of differences algorithm and input two dimensional images.

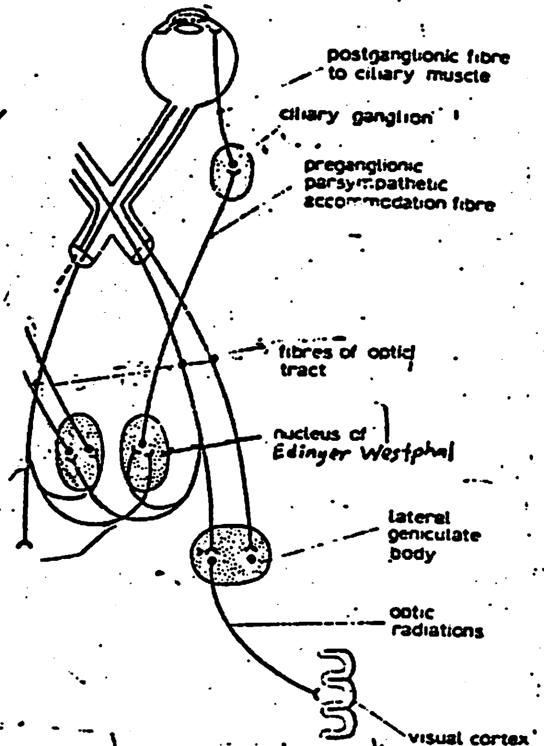
With these simple improvements, the focuser could focus at speeds competitive with the Visitronic and Polaroid system. Since the focusers' hardware is also a video camera, this system is a natural automatic focusing aid choice for video cameras.

The Luminescence Difference Sum bears close resemblance to known processing within animal's visual systems. Within the eye itself are photoreceptors and 3 layers of visual processing(Figure 11). The receptors (rods and cones) synapse (connect) with horizontal cells and bipolar cells. The horizontal cells synapse with more than one receptor and cause the receptive fields of bilateral cells to exhibit the first example of center-surround inhibition (the tendency for stimuli presented around the receptive field of a cell to inhibit its activity) in the visual system. This center-surround inhibition creates edge sensitivity in cell response. amacrine cells synapse with more than one bipolar cells to introduce further laterally sensitive processing. The retinal ganglion cells connect to amacrine and bipolar cells to carry visual information along the visual tract. The sum of the outputs of 2 retinal ganglion cells with complementry center-surround responses correspond is the absolute value of the difference in light level between these 2 fields. The sum of such responses in the foveal area (human area of high resolution vision) is the LDS. It could be maximized over change in accomodation which gives best

Circutry of The Retina and Focus

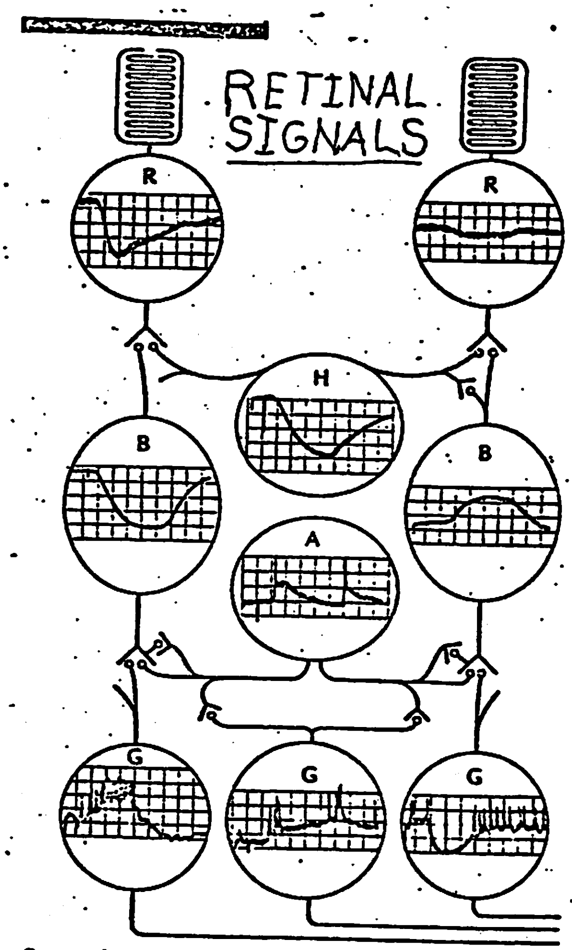
Figure 11

VISUAL PATHWAY



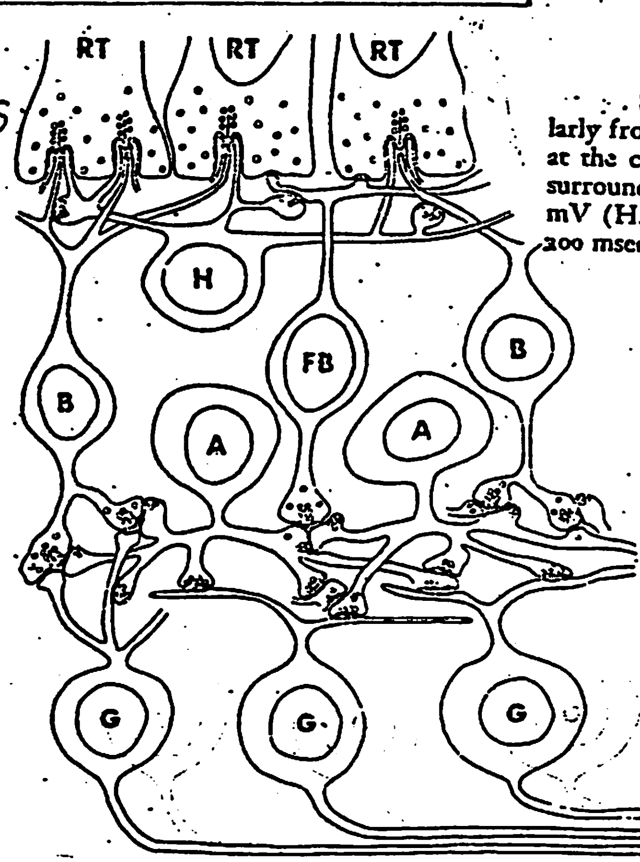
Pathways for the accommodation reflex (Modified after Wyburn, 1960, fig. 71, p. 98)

RETINAL SIGNALS



Synaptic actions in the vertebrate retina, as recorded intracellularly from neurons in *Necturus* (mudpuppy). *Left*, responses recorded at the center of a spot of light (bar above). *Right*, responses in the surround. Voltage calibrations, one scale division equals 1 mV (R); mV (H, B, and G); 5 mV (A). Time calibration, one division equals 200 msec. (From Dowling, 1970, after Werblin and Dowling, 1969.)

RETINAL CONNECTIONS



Vertebrate. A schematic drawing of the arrangements of synaptic contacts found in vertebrate retinas. In the outer plexiform layer, terminal processes of the bipolar cells (B) and horizontal cells (H) penetrate into invaginations in the receptor terminals (RT). The terminals of flat bipolar cells (FB) make superficial contacts on the bases of some receptor terminals. In the inner plexiform layer, two basic synaptic contacts appear to occur. Bipolar terminals may contact one ganglion cell (G) dendrite and one amacrine cell (A) process or bipolar focus. Marked center-surround antagonism is a notable quality of the response characteristics for receptive fields of cells throughout the visual systems pathway.

The visual pathway for focus is very simple (Figure 11). It is possible that the lateral inhibition within the eye is a major part of the processing in most focus decisions. The lens is contracted and expanded to focus. The ciliary muscle which does this is driven by the ciliary ganglion. The cilary ganglion is driven by fibers originating at the Nucleus of Edinger Westphal. The Nucleus of Edinger Westphal is enervated directly through the prectectum from retinal ganglion cells and the superior colliculi (another visual center).

In this way the focus system in animals is fed a signal in differences of light levels with respect to spatial distance. With only a minimum of 4 interaction sites, the information from the retinal ganglion cells focus the eye. Certainly the edge sensitive processing in the eye decreases the work required by focusing circuitry to a bare minimum. For physiological processing to be equivalent to our luminescence Difference Sum (LDS) 2 sums have to be taken. Whether this is actually realized in the way the brain processes information to make accomodation decisions is speculative.

REFERENCES

Boudreau,J.C.,Tsuchitani,C.1973.Sensory Neurophysiology.New York:Van Nostrand Reinhold Ccmpany

Cooper,L.K.,Fulkerson,D.E.and Stauffer,N,L.1978. Linear IC automatically focuses camera lens.Electronics 139-143. April 27

Dowling,J.E.1968 Synaptic Organization Of The Frog Retina: An electorn microscopic analysis comparing the retinas of frogs and primates. Proc.Roy.Soc.B 170: 205-228.

Dowling,J.E.1970. Organization of vertebrate retinas.Invest.Ophth.9:655-680.

Electronics.1978.Polaroid Focuses With Sound Waves.41-42 May 11

Fry,G.A.1955.Blur of The Retinal Image.Columbus:Ohio State University Press

Monnier,M.1968.Function of The Nervous System. New York:Elsevier Publishing Company

Shepherd,G.M.1974.The Synaptic Orgnaization of the Brain. New York: Oxford University Press

APPENDIX

```
;*********************************************************************
;AUTOFOC
; THIS AN AUTO FOCUSING PROGRAM DESIGNED TO BE USED WITH TED SELKER'S
; RETICON ARRAY BASED HORIZONTAL VISUAL INPUT DEVICE AND
; FOCUSING INSTRUMENT.  IT EXPECTS LSB OF PORT:29 MOTOR OUT
;                                              2B IN START PULSE
;                                              23 DATA READY IN
; IN ADDITION  ALL 8 BITS OF              PORT 2A IS DATA INPUT
; THE PROGRAMS REGISTER USAGE IS:A ACCUMULATOR
;                                 B BEGIN OF SCAN, END OF SCAN
;                                 C RESOLUTION OF DIFFERENCES
;                                 DE PAST DATA, WORKSPACE
;                                 HL PRESENT DATA
; THE PROGRAM CALLS MINMAX TO FIND MAXIMUM AND MINIMUM LIGHT LEVELS
; IT USES  THE MINIMUM OF THE ADDRESSES OF THESE TWO DATA POINTS - IT/2
; (ZERO IF NEG) AS  PLACE TO OPEN THE FOCUS WINDOW.  IT USES THE OTHER
; OR 255 IF NECCESSARY AS THE MAXIMUM ELEMENT IN THE FOCUS WINDOW.
;
;*********************************************************************
AUTOFOC 00    00
BEG                                           20B9
END                                           20BA
RES                                           20BB
         CD    CALL               START:CALL ROUTINE TO INITIALIZE SYSTEM
         INIT

         CD    CALL                 INITIALIZE REGISTERS
         REGINIT

         CD    CALL                 CALL ROUTINE TO GET MINIMUM AND MAXIMU
         MINMAX                     IN D AND E

         21    LXI H                GET AT
         MIN                       ADDRESSES IN MIN AND MAX

         7E    MOV A M              FIND BEGINNING AND END OF WINDOW
         73    INC HL              IF ADDRESS OF MAX LIGHT LEVEL IS LARGER
         96    SUB                 THAN MIN'S ADDR (MAXADDR -MINADDR) THEN
         47    MOV B A
         F2    JP
         MINBG

         EE    XRI                    TAKE INVERSE OF DIFFERENCE AND DIVID
         FF    FF
         1F    RAR                    BY 2
         47    MOV B A                SAVE A (MAX -MIN)/2
         7E    MOV A M                PUT MAXADDR IN A
         2B    DEC H L                GET MIN
         4E    MOV C M                     PUT MINADDR IN C
         C3    JMP                  <:ENDIF
         SCNAD

MINBG    1F    RAR                  DIVIDE BY TWO
         47    MOV B A              SAVE (MIN - MAX)/2 IN B
         4E    MOV C M                ELSE PUT MINADDR = ADD OF MAX IN C
         2B    DEC HL
         7E    MOV A M                PUT MAXADDR =ADD OF MIN IN A
```

```
;**********************************************************************
;
; CALCULATE MINIMUM AND MAXIMUM SCAN ADDRESSES
; A HAS MINADDRESS
; B HAS OFFSET
; C HAS MAXADDRESS
;
;**********************************************************************

SCNAD   00     00               MAKE BEGINING OF SCAN-WINDOW
        90     SUB B              TAKE MIN -(MAXADDR - MINADDR)/2
        F2     JP                 IF MIN END IS OFF END OF ARRAY THEN
        STW


        3E     MVI A                BEGINING OF SCAN WINDOW =0
        00     00                     .
STW     21     LXI H              SET 8002 TO MIN OF SCAN
        BEG

        77     MOV M A            OUT IT
        00     00                MAKE END OF SCAN WINDOW
        79     MOV A C            GET MAX= MAX+(MAX-MIN)/2
        80     ADD B
        D2     JNC                IF GREATER THAN SCANLINE LENGTH THEN
        MAXST                           .

        3E     MV A I             SET WINDOW END TO 255 (SCANLINE LENGTH
       FF      FF              .
MAXST   23     INC HL           STORE WINDOW END
        77     MOV M A          .

;**********************************************************************
; MAKE RESOLUTION OF NUMBER OF ELEMENTS TO AVERAGE OVER
;**********************************************************************
        7A     MOV A D           ADDITIVE INVERSE OF 255-MAXLIGHT-MINLIG
        93     SUB E
        06     MVI B
        00     00                 LOOP TO DOIT
INVT    14     INC A                .
        04     INC B                .
        C2     JNZ                ENDLOOP
        INVT

        23     INC H L            STORE THIS CONSTRUED RESOLUTION
        70     MOV M B
        CD     CALL               CALL FOCUS
        FOCUS
```

```
;*****************************************************************************
;FOCUS
;  THIS ROUTINE IS CALLED WITH OPERANDS:
;                  BEG = BEGINING OF SCAN WINDOW
;                  END = END OF SCAN WINDOW
;                  RES = RESOLUTION OF AVERAGING
; IT MAKES A SUM OF THE ABSOLUTE VALUE OF THE DIFFERENCE IN
; LIGHT LEVELS OF AREAS RESCLUTION RETICON ELEMENTS LONG OVER
; THE PORTION OF THE ARRAY ADDRESSED BETWEEN SCAN-BEG AND
; SCAN-END.
;     THIS SUM IS COMPARED TO THE CORRESPONDING SUM TAKEN
; BEFORE  FOCUS MOVEMENT.  IF IT IS SMALLER THAN THE
; LAST SUM IT INDICATES THAT CONTRAST IS GETTING WORSE.
; THE MOTOR DIRECTION IS CHANGED.  IF IT IS BIGGER IT MEANS
; THAT GLOBALLY CONTRASTS ARE GREATER AND FOCUS IS BETTER.
;  PORT 29 =MOTOR
;  PORT 2A =DATA
;  PORT 2B =START PULSE
;  PORT 23 =DATA READY PULSE
; REGISTER USAGE: A = ACCUMULATOR
;                  B = BEGIN,END OF WINDOW TO FOCUS IN
;                  C = RESOLUTION
;                 DE= PAST SUM OF "CONTRASTS"
;                 HL= PRESENT SUM OF "CONTRASTS"
;
;*****************************************************************************
FOCUS    00    00                    START
         CD    CALL                  ZERO REGISTERS
         REGINIT

         E5    PUSH HL              INITIALIZE LAST SUCCESSIVE DIFFERENCES
START    11    LXI DE                INITIALIZE SUM TO ZERO
         00
         00
         D5    PUSH DE
         CD    CALL                 GET START PULSE
         GTSTRT

         21    LXI HL                GET PARAMETERS
         BEG


STRT     46    MOV B M               GET BEGIN
         CD    CALL                  REPEAT UNTIL SCAN WINDOW
         GETRDY

         05    DEC B                  DEC COUNTDOWN
         FA    JP                    END-GET-READY
         RDY

         C3
         STRT                       END WHILE

RDY      23    INC HL                GET END OF WINDOW
         46    MOVB M
         23    INC HL
STPLUS   4E    MOV C M                    GET RESOLUTION
         21    LXI HL
         RESUP
```

```
;*************************************************************************
;
;  READ IN DATA ADDING TO HL RESOLUTION TIMES(THEN SUBTRACT
;  RESOLUTION TIMES)
;
;*************************************************************************
PLUS      00    00                REPEAT IN CONTRAST DIFFS UNTIL LINESIZE
                                   REPEAT IN ADD TO DIFF UNTIL RESOLUTION
          DB    IN                   REPEAT UNTIL READY
          23                           IN READY
          1F    RAL
          D2    JNC                  END-READY
          PLUS

          DB    IN                   READ DATA
          2A                         DATA
          57    MOV D A                  MOVE TO ADD REGISTER
          19    DAD D                ADD TO DOUBLE RET SESUST IN HL
          0D    DCR C              END-REPEAT +DATA
          FA    JM
          NEG

          05    DEC B             END IN ADDR TO DIFF
          F2    JP                  IF END OF WINDOW THEN
          DNE                       .

          C3                       <:EXIT REPEAT-CONTRAST
          PLUS

          0E    LDI C              REPEAT IN-DATA
RESUP     00                       THE INSTRUCTION TO BE MODIFIED
NEGA      05    DEC B                IF NOT END THEN
          FA    JM                   END2 REPEAT CONSTRAST
          DNE   .

NEG       00    00                 REPEAT IN SUBTRACT DATA UNTIL RESOLUTI
          DB    IN                   GET READY
          23
          IF    RAR
          D2    JNC                  END-GET-READY
          NEG

          DB    IN                   IN DATA
          2A    DATA                 .
          EE    XRI                  SUBTRACT FROM SUCCESSIVE SUM
          FF
          57    MOV D A            ADD TWO SUCCESIVE DIFFERENCES
          19    DAD                  .
          0D    DEC C            END-REPEAT-NEG DATES
          F2    JP                 .
          NEGA
```

```
        D5     POP DE              ADD THIS AVERAGED CONTRAST TO PRESENT
        3E     MVI A               SUCCESSIVE DIFFERENCE
        FF     -1                    GET PRESENT SUCCESIVESUM FROM STACK
        BC     CMP H              MAKE THIS AVGD CONT POSITIVE
        D2     JNC                   ABSOLUTE VALUE
        OK

        3E     MVI A                  IF NEEDED INVERT HIGH ORDER BYTE
        FF
        AC     XRA H
OK      3E     MVI A                 SAME FOR LOW ORDER BYTE
        FF     -1
        BD     CMP C
        D2     JNC
        OK2

        3E     MVI A
        FF
        AD     XRA L
OK2     19     DAD D                ADD AS ADVERTISED
        E5     PUSH HL            SAVE
        C3     JMP                  END UNTIL LINESIZE
        STPLUS

;**************************************************************

DNE     00     00                 DECIDE HOW TO IMPROVE FOCUS
        E1     POP HL               GET THIS PASSES SUCCESSIVE DIFFERENCE
        D1     POP DE               GET LAST PASSES SUCESSIVE DIFFERENCE
        E5     PUSH HL              SAVE THIS PASSES SUCCESSIVE DIFFERENCE
        7C     MOV A D                SUBTRACT HIGH ORDER BYTE
        92     SUB H
        C2     JNZ
        NEG2

        7D     MOV A L        SUBTRACT LOW ORDER BYTE
        93     SUB E
NEG2    F2     JP
        END

        CD     CALL
        MOTOR

END     C3     END REPEAT-FOCUS
        START
```

```
;*****************************************************************************
;OLD SIMPLE FOCUS
;     THIS ROUTINE TAKES THE SUM OF SUCESSIVE DIFFERENCES OF LIGHT LEVEL
; ON SUCCESSIVE ELEMENTS OF A 256 ELEMENT ARRAY.  THIS SUM IS COMPARED TO
; SUM TAKEN ON THE LAST ITERATION; TO MAXIMIZE CONTRAST WE ALLOW THE FOCUS
; MOTOR TO CONTINUE IN THE SAME DIRECTION ONLY IF THE NEW SUM IS GREATER T
; THE LAST.
; THE BEGINNING AND END POINTS OF THE SAMPLE ARE USER DEFINED (STORED I
; ATIONS              )
; ROUTINES USED: 2090 ZERO REGISTERS
;                2010 INITIALIZE SYSTEM
;                2875 MOTOR DIRECTION CHANGE
; REGISTER USAGE; A DATA AND COMPUTATION
;                  B +DATA AND START OF SCAN COUNTER
;                  C END OF SCAN COUNTER
;                 DE PRESENT SUCESSIVE DIFFERENCE SUM
;                 HL PAST SUCESSIVE DIFFERENCE SUM
;
;*****************************************************************************
START    00     NOOP              BEGIN

         21     LXI H L            INITIALIZE LAST DATA TO BAD FOCUS
         00     00
         00     00
         E5     PUSH
TOPTP    CD     CALL               CALL INITIALIZE REGISTERS
         REGINIT

         0E     LXI C              LOAD END OF SCAN CONSTANT
         F0     F0                 *IMMEADIATE DATA*
         06     LXI B              LOAD BEGINNING OF SCAN CONSTANT
         01     01                 *IMMEADIATE DATA*
         CD     CALL               GET BEGINNING OF SCAN PULSE
         GTSTRT                     .

FNDBG    05     DEC B              REPEAT UNTIL BEGINNING OF FOCUS AREA
         CA     JZ                   <:EXIT IF DONE
         TOP

RDYBG    CD     CALL               GET READY PULSE
         GTRDY

         C3     JMP                END-REPEAT-BEGIN-OF-SCAN
         FNDBG
```

```
;*************************************************************************
; DATA RETRIEVAL PORTION OF PROGRAM
;*************************************************************************
TOP      00     00                FOCUS
         DB     IN                  REPEAT UNTIL DATAREADY PULSE
         23     READY                 .
         1F     RAR                   .
         D2     JNC                 END-REPEAT-UNTIL-READY
         TOP

         DB     IN                  IN DATA
         2A     DATA
         47     MV B A             PUT A COPY OF THIS ADD DATA IN REG B


MINUS    DB     IN                  REPEAT UNTIL MINUS DATA READY PULSE
         23     READY                 GET READY PULSE
         1F     RAR                   .
         D2     JNC                 END-REPEAT-UNTIL-MINUS
         MINUS

         DB     IN                  GET MINUS DATA
         2A     DATA                .
         90     SUB B               SUBTRACT MINUS FROM PLUS DATA TO GET D
         F2     JP                  IF NEGATIVE THEN
         POS

         EE     XRI                   NEGATE TO MAKE POSITIVE (WE MAKE SUM
         FF     FF                    OF INTENSITIY DIFFERENCES ULTIMATELY
POS      00     00                NOOP
         8B     ADC E             ADD TO SUCESSIVE DIFFERENCES
         5F     MOV E A           THIS, OUR MEGER CONTRIBUTION
         F2     JP               IF SUCCESSIVE DIFFERENCE OVERFLOWS THEN
         NOV

         14     INC D               ADD 1 TO OVERFLOW SUCCESSIVE OVERFLOW
         1E     MVI E              REZERO LOW CRDER BYTE OF SUM
         00
NOV      0D     DCR C             IF NOT END OF FOCUS AREA THEN
         C2     JNZ               END-REPEAT-FOCUS
         TOP
```

```
;*********************************************************************************
; COMPARE SUCCESSIVE DIFFERENCES FROM SUCCESSIVE FOCUS SAMPLES
;*********************************************************************************
        E1     POP H L          GET LASTDATA
        D5     PUSH D E         SAVE NOW DATA
        7C     MV A H
        92     SUB D            SUBTRACT SUCCESSIVE DIFFERENCES
        C2     JNZ              IF ANOTHER BYTE
        NOBTE
        7D     MV A L             GET IT
        93     SUB E              COMPARE IT TOO
NOBTE   F4     CP               IF COMPARE IS PLUS THEN CALL MOTOR CHAN
        MOTOR
        CD     CALL               CALL DELAY PROGRAM TO ADD HYSTERISIS
        DELAY

        C3     JMP              END-REPEAT-FOCUS
        TOPTP
```

```
;***************************************************************************
;INIT
;      INITIIALIZES HARDWARE AND SOBTWARE OF THE SELKER FOCUS SYSTEM;
;  STACKPOINTER MUST BE PRESET BY USER TO 20C8 FOR THIS ROUTINE TO
;  RETURN CORRECTLY.
;***************************************************************************
INIT        3E     MVIA            INITIALIZE MOTOR TO STOPED
            00
            32     STAI            .
            BD                     MOTOR DIRECTION IS IN 20BD
            20
            3E     MOVIA           SET INTERUPT MASK
            08
            30     SIM             TURNEM ON
            FB     EI
            3E     MVIA            INITITIALIZE BASIC RAM IO
            03                     21,22 ANOLUGUE OUT, 23 DATAREADY IN
            D3     OUT
            20
            3E     MVIA               INITIALIZE EXPANSION RAM IO
            01                     29 MOTOR OUT,2A IN DATA,2B IN STARTPULSE
            D3
            28
END         C9     RET
```

```
;*********************************************************************
;GTSRT
;  THIS PROGRAM RETURNS TO USER WHEN RETICON GIVES A START PULSE
;
;*********************************************************************
GTSTRT   DB    IN                FIND START PULSE
         2B    START
         1F    RAR                 SHIFT IT INTO CARRY
         D2    JNC               END-FIND-LOOP
         GTSTRT

END      C9    RET
;***************************************************************
;GTRDY
;    THIS PROGRAM LOOPS PROBING FOR A DATAREADY SIGNAL FROM THE
; THE A TO D CONVERTOR.
;
;***************************************************************
GTRDY    DB    IN                REPEAT UNTIL READY PULSE
         23    IN IT
         1F    RAR                 SHIFT IT INTO CARRY
         D2    JNC                END-REPEAT
         GTRDY

END      C9    RETURN WHEN RDY
;**********************************************************
;REGINIT
;  THIS ROUTINE ZEROS REGISTIERS A,B,C,D AND E
;
;**********************************************************
REGININT 3E    MVAI              ZERO A
         00
         47    MOV B A
         4F    MOV C A
         57    MOV D A
         5F    MOV E A
         67    MOV H A
         6F    MOV L A
END      C9    RET
```

```
;
;
;*******************************************************************
;MOTOR
;     THIS PROGRAM CHANGES THE DIRECTION OF THE MOTOR.
;   TO MAKE THE MOTOR GO FORWARD IT SETS PORT 29 TO 01 TO MAKE
;   THE MOTOR BACK UP IT SETS PORT 29 TO 10 (LOW ORDER 2 BITS).
;   IT STORES MOTOR DIRECTION IN LOCATION MOTOR(20BD).
;
;*******************************************************************
MOTADD                                        20BD
MOTOR     3A     LDAI                    GET MOTOR DIRECTION
          MOTADD


          EE     XRI                     CHANGE ITS DIRECTION
          03
          D3     OUT                     OUT IT TO MOTOR
          29
          32     STAI                    STORE NEW DIRECION
          MOTADD

END       C9     RET
0;*******************************************************************
;END
;      THIS ROUTINE CHECKS IF THE FOCUS MOTOR HAS COME TO THE END OF ITS
; TRACK AND PUSHED A SWITCH TO MAKE THE FIRST BIT OF PORT 21 GO HIGH.
; IF THE MOTOR HAS COME TO ITS END, THE MOTOR DIRECTION CHANGE ROUTINE
; IS CALLED.
;
;*******************************************************************
END       DB     IN                      BEGIN GET END SIGNAL
          21                               FROM PORT 21
          1F     RAR
          DC     CC                        IF END OF TRACK CALL MOTOR CHANGE ROUTINE
          MOTOR

          C9     RET                     END
```

```
;*************************************************************************
;MINMAX
;     THIS ROUTINE SCANS THE RETICON'S ELEMENTS TO FIND THE BRIGHTEST
; & DULLEST PLACES.  IT RETURNS MINIMUN BRIGHTNESS LEVEL IN REGISTER D
;                                 MAXIMUM BRIGHTNESS LEVEL IN REGISTER E
; MINIMUM BRIGHTNESS ELEMENT'S ADDRESS IN LOCATION MIN (20C0),
; MAXIMUM BRIGHTNESS ELEMENT'S ADDRESS IN LOCATION MAX (MIN+1)
;
; INTERNALLY, REGISTER B IS USED FOR NEW DATA
;             REGISTER C IS USED FOR NEW ADDRESSE
;
;*************************************************************************
MINMAX   00     00                START
         21     LXI HL            SET UP FOR MEMORY ACCESS OF MIN AND MAX
         C0
         20

         11     LX1 DE            INIITIALIZE MIN=FF, MAX =0
         00
         7F
         0E     MVI C               ZERO ELEMENT COUNTER
         00     00
         CD     CALL                GET START OF SCAN PULSE
         GTSTRT

NOTHER   00     00                  REPEAT UNTIL END OF ARRAY
         CD     CALL                  GET READY PULSE
         GTRDY

         DB     IN                    GET ELEMENTS VALUE
         2A     DATA
         47     MOV B A               SAVE
         92     SUB D                 IF SMALLEST YET THEN (SUBTRACT MIN)
         E2     JP                    .
         BIGR                         .

         50     MOV D B                 MIN=NEW DATA

         71     MOV M C               SAVE ADDRESS
         78     MOV A B
         C3     JMP
         ENDLP

BIGR     93     SUB E                 IF ACC BIGGER THEN
         FA     JM
         ENDLP

         58     MOV E B                  MAX= NEW DATA
         23     INC HL                  SAVE ADDRESS
         71     MOV M C                 IN MAX
         2B     DEC HL
ENDLP    0C     INC C                 INCREMENT ELEMENT COUNTER
         C2     JNZ                 END-LOOP
         NOTHER

END      C9     RET               DONE
```

```
;*******************************************************************
;READEM
;     THIS ROUTINE LETS A USER INPUT A NUMBER WHICH IS USED AS THE
; ADDRESS OF A LOCATION IN THE RETICON TO POLL.  THE LOCATION'S
; CONTENTS ARE DISPLAYED AND UPDATED CONTINUEOUSLY.
;
;*******************************************************************
READEM    00     00                 START
          CD     CALL                 GET USER INPUT
          E7
          02
          47     MOV B A              PUT USERS ELEMENT INPUT
          4F     MOV C A              IN REGISTERS B AND C
R EAD       CD     CALL                 GET START PULSE
          GTSTRT

ELEMNT    CD     CALL                 REPEAT UNTIL USER SPECIFIED ELEMENT
          GTRDY                         GET READY PULSE

          05     DCR B                COUNT DOWN B TO USER SPECIFIED ELEMENT
          C2     JNZ                  END-REPEATUNTIL ELEMENT
          ELEMNT

          DB     IN                   READ DATA
          2A
          CD     CALL                 DISPLAY DATA
          6E
          03
          41     MOV B C              UPDATE CCUNTER
          C3     JMP                END
          READ
;*******************************************************************
;MOTDEM
;     THIS PROGRAM LETS THE USER INPUT A NUMBER WHICH IS OUTPUTED TO
; THE MOTOR
;
;*******************************************************************
MOTDEM    CD     CALL               START GET USER DATA
          E7                          CALL READ KEYBOARD
          02
          CD     CALL                DISPLAY USER INPUT
          6E                        CALL DISPLAY ROUTINE
          03
          D3                         OUT DATA TO MOTOR
          29
          C3
MOTDEM
```

```
;*****************************************************************
;HISTO
;    THIS PROGRAM HISTOGRAMS DATA IN LOCATIONS 2800-28FF
; IT OUTPUTS X'S AT PORT 21 AND Y'S ATR 22
;
;*****************************************************************
HISTO   21   LXI HL          SET MEMOR READOUT AREA
        FF
        28
HIST    7E   MOV A M         GET PIECE OF DATA
        D3   OUT
        22
        7D   MOV A L         GET X
        D3   OUT
        21
        2D   DEC L
        C2   JNZ             END LOOP
        HIST

        C3   JMP             END HISTO
        HISTO
;**************************************************************
;STBUFF
;     THIS ROUTINE STORES A COPY OF THE BRIGHTNESSES OF ALL
; ELEMENTS OF THE RETICON AT LOCATIONS 2800-28FF.
;
;**************************************************************
STBUFF  00   00              START
        CD   CALL             INITIALIZE REGISTERS TO ZERO
        REGINIT

        21   LXI              INITIALIZE MEMORY TO WRITE TO
        00
        28
        CD   CALL              GET START PULSE
        GTSTRT
ILL     CD   CALL              GET READY PULSE

        GTRDY

        DB   IN                READ DATA
        2A
        77   MOV M A          STORE RESLT
        23   INX H L           INC MEMORY
        3E   MOVI A            IF END OF SCAN
        FF
        95   SUBT L
        C2   JNZ
        FILL


        C9   RET
```

```
;**********************************************************************
;NEW SIMPLE FOCUS
:     THIS ROUTINE TAKES THE SUM OF SUCESSIVE DIFFERENCES OF THE
; SQUARES OF THE LIGHT LEVELS ON SUCCESSIVE
; ELEMENTS OF A 256 ELEMENT ARRAY.  THIS SUM IS COMPARED TO THE
; SUM TAKEN ON THE LAST ITERATION; TO MAXIMIZE CONTRAST WE ALLOW THE FO
; MOTOR TO CONTINUE IN THE SAME DIRECTION ONLY IF THE NEW SUM IS GREATE
; THE LAST.
; THE BEGINNING AND END POINTS OF THE SAMPLE ARE USER DEFINED (STORED I
; ATIONS                )
; ROUTINES USED: 2090 ZERO REGISTERS
:                2010 INITIALIZE SYSTEM
:                2875 MOTOR DIRECTION CHANGE
; REGISTER USEAGE; A DATA AND COMPUTATION
:                 B +DATA AND START OF SCAN COUNTER
:                 C END OF SCAN COUNTER
:                DE PRESENT SUCESSIVE DIFFERENCE SUM
:                HL PAST SUCESSIVE DIFFERENCE SUM
:
;**********************************************************************
START   00    NOOP           BEGIN

        21    LXI H L         INITIALIZE LAST DATA TO BAD FOCUS
        00    00
        00    00
        E5    PUSH
TOPTP   CD    CALL            CALL INITIALIZE REGISTERS
        REGINIT

        0E    LXI C           LOAD END OF SCAN CONSTANT
        F0    F0              *IMMEADIATE DATA*
        06    LXI B           LOAD BEGINNING OF SCAN CONSTANT
        01    01              *IMMEADIATE DATA*
        CD    CALL            GET BEGINNING OF SCAN PULSE
        GTSTRT                 .

FNDBG   05    DEC B           REPEAT UNTIL BEGINNING OF FOCUS AREA
        CA    JZ                <:EXIT IF DONE
        TOP

RDYBG   CD    CALL            GET READY PULSE
        GTRDY

        C3    JMP             END-REPEAT-BEGIN-OF-SCAN
        FNDBG
```

```
;*********************************************************************
; DATA RETRIEVAL PORTION OF PROGRAM
;*********************************************************************
        CD     CALL            GET FIRST ELEMENTS VALUE
        GTRDY                      GET READY PULSE

        DB     IN                GET DATA
        2A
        47     MV B A            SAVE IT

TOP     CD     CALL            GET DATA
        GTRDY                    GET READY PULSE
        DB     IN                GET DATA
        2A
        67     MV H A           SAVE DATA
        90     SUB B            TAKE DIFFERENCE
        44     MV B H
        F2     JP                 IF NEGATIVE THEN
        POS

        EE     XRI                  NEGATE TO MAKE POSITIVE (WE MAKE SUM
        FF     FF                   OF INTENSITIY DIFFERENCES ULTIMATELY
POS     00     00              NOOP
        8B     ADC E           ADD TO SUCESSIVE DIFFERENCES
        5F     MOV E A         THIS, OUR MEGER CONTRIBUTION
        F2     JP             IF SUCCESSIVE DIFFERENCE OVERFLOWS THEN
        NOV

        14     INC D              ADD 1 TO OVERFLOW SUCCESSIVE OVERFLOW
        1E     MVI E           REZERO LOW ORDER BYTE
        00
NOV     0D     DCR C           IF NOT END OF FOCUS AREA THEN
        C2     JNZ             END-REPEAT-FOCUS
        TOP
```

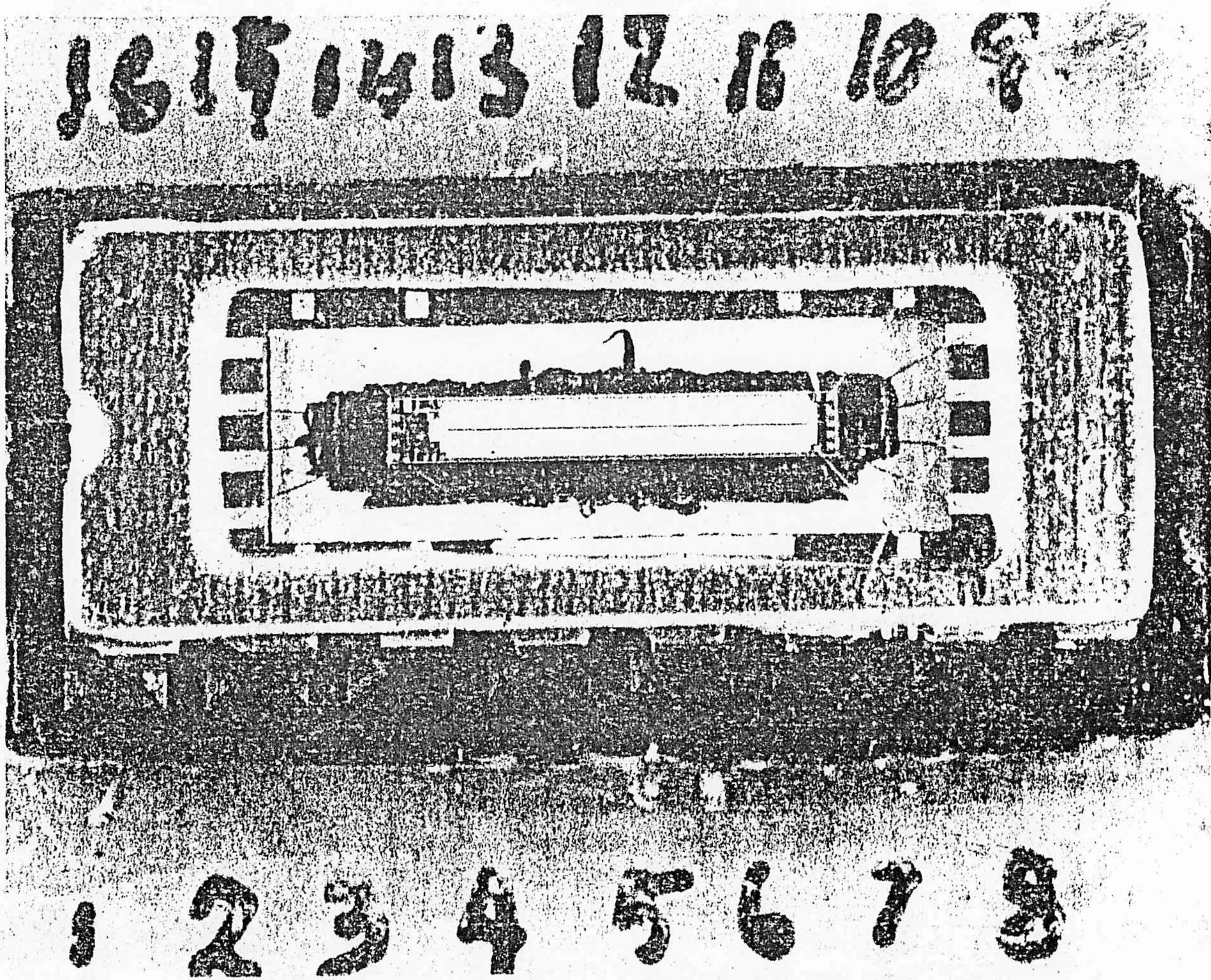
```
;*********************************************************************
; COMPARE SUCCESSIVE DIFFERENCES FROM SUCCESSIVE FOCUS SAMPLES
;*********************************************************************
        E1     POP H L          GET LASTDATA
        D5     PUSH D E         SAVE NOW DATA
        7C     MV A H
        92     SUB D            SUBTRACT SUCCESSIVE DIFFERENCES
        C2     JNZ              IF ANOTHER BYTE
        NOBTE
        7D     MV A L             GET IT
        93     SUB E              COMPARE IT TOO
NOBTE   F4     CP               IF COMPARE IS PLUS THEN CALL MOTOR CHAN
        MOTOR

        CD     CALL               CALL PRGRAM TO INTRODUCE HYSTERYSIS
        DELAY
        C3     JMP              END-REPEAT-FOCUS
        TOPTP
```

```
;*********************************************************************
;FOCUS TO USE SQUARED TABLE
;     THIS ROUTINE TAKES THE SUM OF SUCESSIVE DIFFERENCES OF THE
; SQUARES OF THE LIGHT LEVELS ON SUCCESSIVE
; ELEMENTS OF A 256 ELEMENT ARRAY.  THIS SUM IS COMPARED TO THE
; SUM TAKEN ON THE LAST ITERATION; TO MAXIMIZE CONTRAST WE ALLOW THE FO
; MOTOR TO CONTINUE IN THE SAME DIRECTION ONLY IF THE NEW SUM IS GREATE
; THE LAST.
; THE BEGINNING AND END POINTS OF THE SAMPLE ARE USER DEFINED (STORED I
; ATIONS          )
; ROUTINES USED: 2090 ZERO REGISTERS
;                2010 INITIALIZE SYSTEM
;                2875 MOTOR DIRECTION CHANGE
; REGISTER USEAGE; A DATA AND COMPUTATION
;                  B +DATA AND START OF SCAN COUNTER
;                  C END OF SCAN COUNTER
;                 DE PRESENT SUCESSIVE DIFFERENCE SUM
;                 HL PAST SUCESSIVE DIFFERENCE SUM
;
;*********************************************************************
START    00     NOOP                BEGIN

         21     LXI H L              INITIALIZE LAST DATA TO BAD FOCUS
         00     00
         00     00
         E5     PUSH
TOPTP    CD     CALL                 CALL INITIALIZE REGISTERS
         REGINIT

         0E     LXI C                LOAD END OF SCAN CONSTANT
         F0     F0                   *IMMEADIATE DATA*
         06     LXI B                LOAD BEGINNING OF SCAN CONSTANT
         01     01                   *IMMEADIATE DATA*
         CD     CALL                 GET BEGINNING OF SCAN PULSE
         GTSTRT

FNDBG    05     DEC B                REPEAT UNTIL BEGINNING OF FOCUS AREA
         CA     JZ                     <:EXIT IF DONE
         TOP

RDYBG    CD     CALL                 GET READY PULSE
         GTRDY

         C3     JMP                  END-REPEAT-BEGIN-OF-SCAN
         FNDBG
```

```
;*****************************************************************
; DATA RETRIEVAL PORTION OF PROGRAM
;*****************************************************************
        CD      CALL            GET FIRST ELEMENTS VALUE
        GTRDY                        GET READY PULSE

        DB      IN                GET DATA
        2A
        47      MV B A


TOP     CD      CALL            GET DATA
        GTRDY                     GET READY PULSE
        DB      IN                GET DATA
        2A
        67      MV H A
        90      SUB B            TAKE DIFFERENCE
        44      MV B H
        F2      JP                IF NEGATIVE THEN
        POS

        EE      XRI                 NEGATE TO MAKE POSITIVE (WE MAKE SUM
        FF      FF                  OF INTENSITIY DIFFERENCES ULTIMATELY
POS     00      00              NOOP
        CD      CALL              SQUARE DIFFERENCE
        SQUR
        8B      ADC E           ADD TO SUCESSIVE DIFFERENCES
        5F      MOV E A         THIS, OUR MEGER CONTRIBUTION
        F2      JP             IF SUCCESSIVE DIFFERENCE OVERFLOWS THEN
        NOV

        14      INC D             ADD 1 TO OVERFLOW SUCCESSIVE OVERFLOW
NOV     0D      DCR C           IF NOT END OF FOCUS AREA THEN
        C2      JNZ             END-REPEAT-FOCUS
        TOP
```

```
;*************************************************************************
; COMPARE SUCCESSIVE DIFFERENCES FROM SUCCESSIVE FOCUS SAMPLES
;*************************************************************************
         E1    POP H L          GET LASTDATA
         D5    PUSH D E         SAVE NOW DATA
         7C    MV A H
         92    SUB D            SUBTRACT SUCCESSIVE DIFFERENCES
         C2    JNZ              IF ANOTHER BYTE
         NOBTE
         7D    MV A L             GET IT
         93    SUB E              COMPARE IT TOO
NOBTE    FC    CM               IF COMPARE IS MINUS THEN CALL MOTOR CHAN
         MOTOR

         C3    JMP              END-REPEAT-FOCUS
         TOPTP
```

```
*************************************************************************
* DELAY                                                                 *
* THIS ROUTINE IS USED BY FOCUS PROGRAMS TO DELAY BETWEEN FOCUS         *
* DECISIONS INVERSELY PROPORTIONALLY TO THE FOCUS CONSTANT THE          *
* FOCUS PROGRAM HAS AS ITS MEASURE OF FOCUS.  IN THIS WAY THE           *
* PROGRAM FOCUSES FARTHER BETWEEN REEVALUATIONS OF FOCUS DIRECTION      *
* WHEN IT IS FAR AWAY FROM FOCUS THAN WHEN IT IS CLOSE TO BEST FOCUS.   *
*                                                                       *
*************************************************************************
DELAY    00     NOOP            BEGIN
         00     NOOP
         00     NOOP
         D5     PUSH DE          SAVE FOCUS LDQ
         7B     MV A E           DISPLAY LDQ
         CD     CALL
         6E
         03
         D1     POP DE           RESTORE FOCUS LDQ
         00     NOOP
         4A     MV C D            GET LOOP CONSTANT= HIGH ORDER OF LDQ{
OLOOP    7B     MV AE            REPEAT UNTIL HIGH ORDER LDQ
ILOOP    3C     INC A              REPEAT UNTIL LOW ORDER LDQ
         C2     JNZ                ENDLOOPI
         ILOOP

         OC     INC C
         C2     JNZ               ENDLOOP
         OLOOP
         C9     RET             END
```


16 15 14 13 12 11 10 9
1 2 3 4 5 6 7 8