5-1-1994

Evaluation of a cooled CID camera system (SICam - The CID-based scientific camera) for low light imaging applications

Chen Tang

EVALUATION OF A COOLED CID CAMERA SYSTEM

(SICam - The CID-based Scientific Camera)

FOR

LOW LIGHT IMAGING APPLICATIONS

by

Chen Tang
B. S. Fudan University
(1987)

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Center for Imaging Science
in the College of Imaging Arts and Sciences of
the Rochester Institute of Technology

May 1994

Signature of the Author_____
Center for Imaging Science

Accepted by Dana G. Marsh June 23, 1994
Coordinator, M.S. Degree Program

Center for Imaging Science
College of Imaging Arts and Sciences
Rochester Institute of Technology
Rochester, New York

CERTIFICATE OF APPROVAL

M. S. DEGREE THESIS

The M. S. Degree Thesis of Chen Tang has been examined and approved by the thesis committee as satisfactory for the thesis requirement for the Master of Science degree

Dr. Z. Ninkov, Thesis Advisor	Date 6/3/94
Dr. R. Easton, Jr.	Date 6/3/94
Dr. D. Meisel	Date 6/3/94

THESIS RELEASE PERMISSION FORM

ROCHESTER INSTITUTE OF TECHNOLOGY

COLLEGE OF IMAGING ARTS AND SCIENCES

CENTER FOR IMAGING SCIENCE

Title of Thesis

EVALUATION OF A COOLED CID CAMERA SYSTEM
(SICam - The CID-based Scientific Camera)
FOR
LOW LIGHT IMAGING APPLICATIONS

I, Chen Tang, hereby grant permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part.
Any reproduction will not be for commercial use or profit.

Signature:_____

Date: June 3, 1994

EVALUATION OF A COOLED CID CAMERA SYSTEM

(SICam - The CID-based Scientific Camera)

FOR

LOW LIGHT IMAGING APPLICATIONS

by

Chen Tang

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Center for Imaging Science
in the College of Imaging Arts and Sciences of
the Rochester Institute of Technology

ABSTRACT

The performance of liquid nitrogen (LN_2) cooled digital scientific instrumentation camera incorporating a large format (512 x 512, 20.3 mm diagonal) Charge Injection Device (CID) imager which was fabricated by CIDTEC® for use in low-light imaging applications is

described in this thesis. The LN_2 cooled camera outputs 14-bit digital pixel data which can be read out randomly and non-destructively from individual pixels, multiple sub-arrays, or from the entire imager. CID's have the unique capability to skim accumulated pixel charge and/or non-destructively read pixel information, enabling the system user to monitor events and dynamically adapt application exposure in real-time for the entire array or individual pixels. Non-destructive read-out also is used to extend linear dynamic range. The Intel® 80286 based camera system operates from PC/AT compatible platforms through a digital I/O board.

Topics in this thesis include the CID camera system overview, mechanical and electronic components, dewar cooling, system calibration, programming of the driver software and final performance evaluation of the system.

ACKNOWLEDGMENTS

The assistance of the following people was greatly appreciated and essential to the successful completion of this thesis:

Dr. Z. Ninkov, thesis advisor, who was always available and willing to provide assistance and advice whenever needed;

Dr. R. Easton, thesis committee member, for his valuable comments and material support.

Dr. David D. Meisel, thesis committee member, for his mental support and valuable comments.

Billy Leung, for his help on programming and understanding.

Brian Backer, William Cirillo, Hans Deeg, for their valuable comments and help during the experiments. This thesis could not be done without those lunches at Ritz and subs.

Special thanks to my wife Yang for her understanding and supports in all the aspects.

DEDICATION

This thesis is dedicated to

my parents

and

my beautiful wife Yang.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

1. Background

The Charge Injection Device (CID) is an X-Y addressable two-dimensional array of photosensitive MOS capacitor elements invented by the General Electric Company (GE®) in 1972[18]. The unique pixel structure of the CID results in significant benefits in performance. The layout of a typical CID pixel is shown in figure 1. A pixel is formed by two intersecting MOS capacitors which are formed by polysilicon over gate oxide. If the two capacitors are laid out orthogonally, so the polysilicon of one capacitor joins with that of the adjacent pixel, a continuous run of capacitors is formed which may be used to control the pixels from the border of the array. The vertical polysilicon run is

called the column electrode and the horizontal polysilicon run is called the row electrode. In the typical CID architecture, where the charge is sensed across the row, a metal strap is connected parallel to the row electrode. This is necessary to reduce noise and improve the RC time constant. The metal strap is the only photo insensitive structure on the pixel and it typically only 2 - 2.5 microns wide.

Figure 1 shows the pixel layout of CID 38 imager (512 x 512, 20.3 mm diagonal) which was manufactured by CIDTEC® for use in the scientific instrumentation camera system called SICam™.

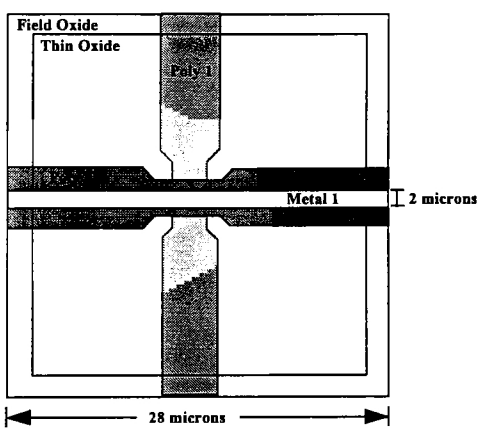


Figure 1 : CID 38 pixel layout

Approximately 60% of the active area of the CID imager is exposed gate oxide that provides broad spectral response. It can be seen that the metal strap runs along the row electrode. The metal run is in direct contact to the second polysilicon layer. To further enhance spectral response, the second polysilicon layer is approximately 800 Å thick. The crossover of the polysilicon electrodes is a major source for pattern noise, hence they have been narrowed to minimize the effect[4].

Integrating a scene on a CID involves three stages in the read-out process (Figure 2). At the beginning of an integration time, the charges (holes) are injected into the substrate material to neutralize (combine holes with electrons) the present charge. Incident photons generate holes which are accumulated under a negatively biased column storage pad. At the conclusion of an exposure, the signal charge stored under a pixel located at a selected row and column is transferred

within that pixel to the row pad. The row pad is attached to a row preamplifier and this signal is output from the chip.

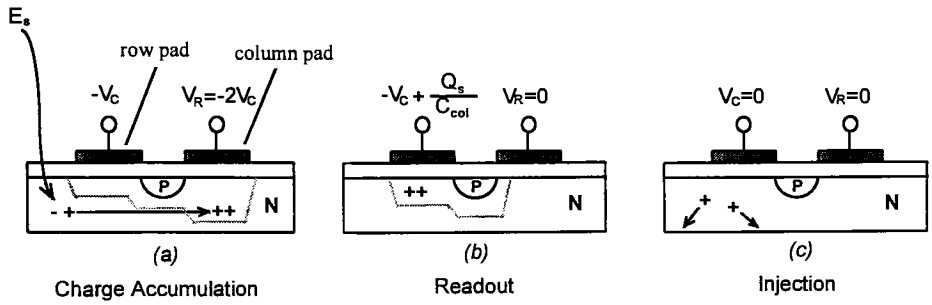


Figure 2 : CID Integration Sequence.

2. CID features

CID imagers are two-dimensional X-Y addressable MOS arrays fabricated over an underlying charge collection layer. Each element being comprised of two MOS charge storage capacitors per pixel. Adjacent pixels are electronically isolated from one another by an implant supplemented field oxide. This structure is described in the literature[5] and provides unique

operating properties as well as performance features that can directly ease or enhance solutions to challenging application problems. Unique performance features of CID construction include non-destructive pixel read-out (NDRO), exceptional radiation tolerance, inherent anti-blooming characteristics, and true random pixel addressibility.

2.1. Non-destructive read-out (NDRO)

During read-out of CCD devices, collected charge is transferred from the pixel sites to an output port on the device periphery, taking the image information out of the array. Read-out is "destructive" since the image stored on the sensor is effectively "erased" as it is read out. Each pixel must integrate charge and be read again to find the proper pixel exposure for the scene.

Read-out of CID devices is accomplished by transferring and sensing the integrated charge packet

within the individual pixel site and not necessarily discharged. Read-out is "non-destructive" because charge remains intact in the pixel after the signal level has been determined. Charge injection, ("clearing the pixel") is a separate, controllable operation.

NDRO is the unique ability of CID devices to read out the signal level of a selected pixel without altering the charge and without removing it from the respective pixel site. The following diagram shows the pixel operation of NDRO.

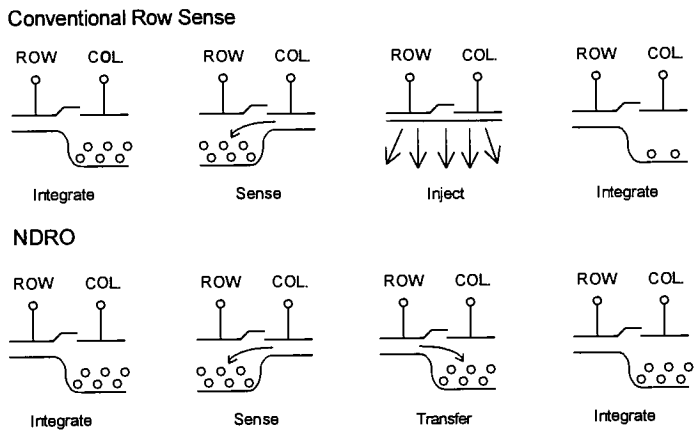


Figure 3 : Pixel Operation of NDRO.

This capability provides the user with three powerful tools. First, during image acquisitions

requiring lengthy integration times, image exposure can be monitored and dynamically adapted in real-time. Second, moving objects can be tracked by read-out of selected pixels and/or sub-arrays. Third, because CID read-out noise is Gaussian, pixel read noise can be reduced by the square-root of the total number of read-outs. Together, these features permit the user to fully monitor, assess and control the data collection, as well as to extend the operating dynamic range of the sensor[10].

2.2 Anti-blooming

CID imagers offer intrinsic anti-blooming performance even under intensely concentrated illumination. The CID is fabricated on an underlying N^+ epitaxial layer over P-channel substrate. Charges which cannot be contained within the pixel well (overload), diffuse vertically into the substrate where they recombine instead of diffusing into adjacent pixels.

Such diffusion would cause image smear or channel blooming. The collected charge never leaves the pixel site during integration; the signal continues to integrate until the pixel reaches saturation. At saturation, the pixel can no longer retain signal charge and additional charges are swept into the reversed biased substrate. The substrate functions as a buried collector and since it occupies the entire imager, there is no lateral path for signal charge. The CID allows high-contrast exposure and read-out of adjacent pixels, thus confining overloads to affected pixels.

2.3 Radiation tolerance

The CID offers radiation resistance to levels exceeding 10^6 gamma total dose[14], approximately two orders of magnitude greater than levels achieved with CCD devices. CID's are fabricated using

high-resistivity P-channel process technology. When irradiated by nuclear particles, X-radiation or an electron beam, device channel and gate thresholds increase instead of decreasing as with N-channel CCD's. Consequently, radiation generally causes most CCD's to eventually turn fully "on" or "short circuit", whereas CID's continue to function with a slight degradation in pixel full well capacity. Additionally, while some CCD's function for a limited time in a radiation environment, radiation-induced shot-noise severely limits dynamic range, degrading performance to an unusable level prior to actual malfunctioning. The large pixel storage capacity of the CID permits quantum noise-limited operation, thus lessening the effect of radiation- induced noise.

2.4 Random pixel addressibility

A CID array is formed when column pads of all pixels in a column are connected to a column bus. Similarly, the row pads of all pixels in a row are

connected to form a row bus. Charge can be manipulated within a given pixel by selectively driving the appropriate row and column bus. Row and column selection is accomplished by using shift registers and addressable decoders and multiplexers (switches) mounted orthogonally on the array periphery (Figure 4).

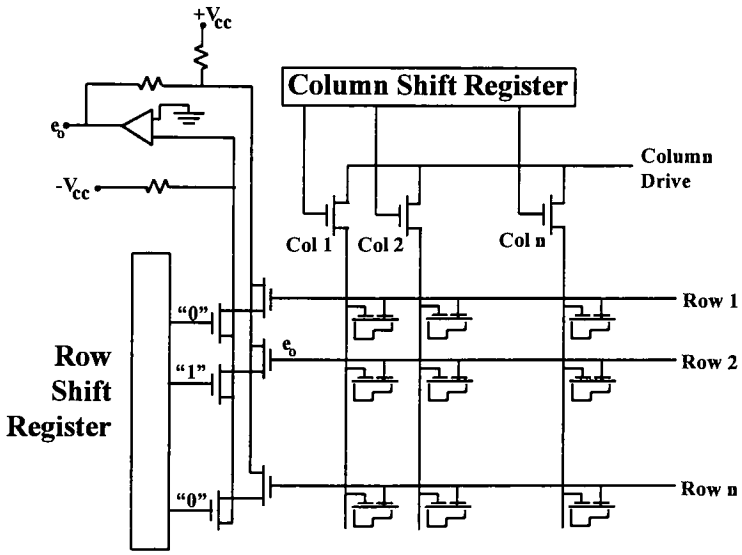


Figure 4 : Pre-amp per row architecture of CID imager.

With row read-out operation, the column decoder selects a column bus and connects it to a drive bus, and the row decoder connects the selected row to a read-out bus. When a drive signal is applied to the

drive bus, signal charge sorted under a pixel located at the coordinates of the selected column and row is transferred within that pixel and sensed as a current or voltage on the read-out bus. In column read-out, the terms row and column in the above operation are simply reversed and signal is sensed on the column bus.

Selective image data acquisition is achieved at variable frame rates. During read-out, selected pixels may be addressed and read out without altering the developing charge levels achieved and without affecting other pixels not intended to be read out at that time. This feature is particularly useful for monitoring key image features during the integration process prior to read-out of the whole image.

II. System Overview

Performance results reported in this thesis have been obtained using a microprocessor-based (Intel® 80286), 14 bit, digital scientific instrumentation camera, called the SICam™, developed by CIDTEC® to take advantage of the full scope of CID properties and capabilities[3]. Figure 5 shows the system block diagram which details a typical camera configuration and the major component blocks employed in the SICam™ system.

1. Description - System Block Diagram

SICam™ is a wide dynamic range (Appendix C) camera system designed to use the full performance potential of the CID imager in scientific imaging applications. To support these different applications, the system was

designed around a programmable architecture rather than on a conventional, dedicated hardware/ firmware platform. The host computer system is a IBM PC with an Intel® 486DX33 CPU, SVGA graphic display, 16MB RAM and 440MB of hard disk space for image storage. The "C" language software driver performs low-level SICam command operations as well as host computer configuration of Direct Memory Access (DMA) channels, to support background transfer of pixel data directly to host computer memory.

The Camera Control Module (CCM) is a macro-programmable command processor providing high-speed signal timing generation, analog signal processing and digitization. It is partitioned into the "286" Processor and Sensor Control sub-systems. Sensor Control receives and interprets "primitive" commands sent from the Processor and performs corresponding state timing sequences or reference level control/monitoring functions. Primitive commands are received from the Processor and stacked through a

single-level pipe-line register. The commands are reduced to binary level count, then enable operations to implement pre-defined state timing engines a specified number of times within the logic configurable array (LCA). Local camera control software interprets high-level commands received from the host computer. The CCM performs preprogrammed functions based on previously loaded software from EPROM, or from the software downloaded by the host computer. Upon receipt of command data, the CCM generates high-speed timing signals required to perform selected CID imager operations. The analog video signal from the Focal Plane Array (FPA) is digitized in the CCM using a Correlated Double-Sample (CDS) method, and transferred in parallel format directly to host computer memory by DMA[3]. All CCM software can be designed in-system on the host computer and downloaded to the camera, enabling execution of custom commands and timing algorithms.

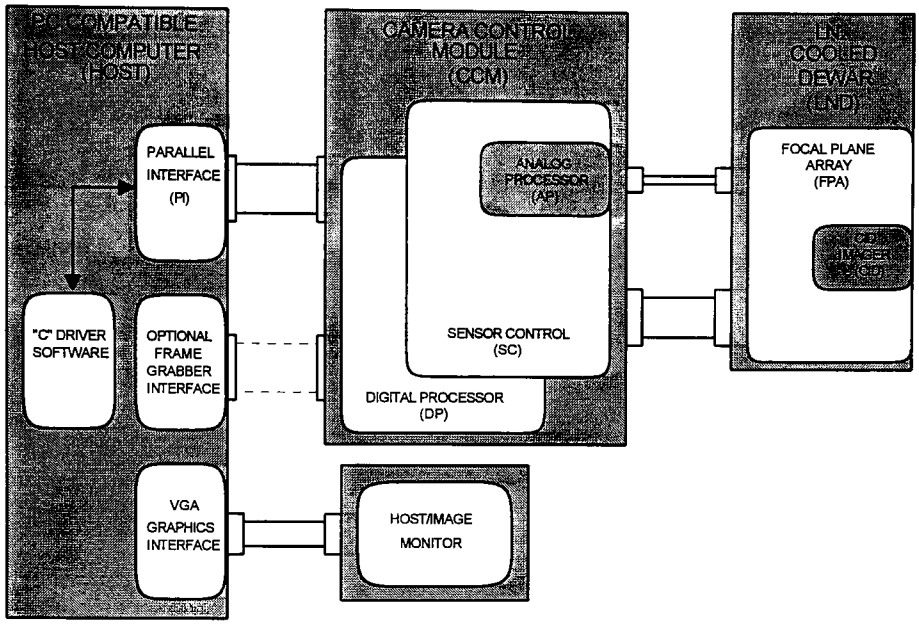


Figure 5 : SICam™: System Block Diagram consisting of a host computer system, camera control module, and LN_2 cooled remote head

Attached to the LN_2 Cooled Dewar are the low noise, large format CID imager, and associated drive electronics and pixel data amplifiers. LN_2 cooling of the CID in a vacuum environment effectively eliminates dark-current noise contributions during extended integration. Pre-amp Per Row amplifiers on the CID imager and video amplifiers on the FPA board provide

analog signal buffering prior to transmission to the camera control module. The post-amplifier amplifies pixel data to an amplitude high enough that it can be transmitted without degradation to the CCM. The FPA assembly is designed for mounting within a dewar and operation at liquid nitrogen temperatures.

2. Features

The main features of the SICam hardware and software interface are summarized below:

- Pipe-lined HOST Command Architecture:
  - minimizes latency between receipt of commands and execution.
- Parallel Command Processing:
  - a local microprocessor and high-speed timing generator work in parallel to enable simultaneous interpretation of commands and generation of timing signals.
- Programmable Architecture:

- command software may be modified using the host computer and Microsoft "C" compiler.
- high-speed timing software may be modified using the host computer and LCA development software,
- command and control I/O functions, or high-speed imager control timing may be modified by the user,
- programmable CID reference voltages enable the use of customized control and calibration functions,
- external I/O available to implement control algorithms for external shutters, strobes, etc. required by the application. Image acquisition may be triggered by external events.

- Precise Data Acquisition:
  - low-noise CID imager and analog processor support wide dynamic range measurements.
- Digital Video Interface:
  - 16-bit digital data output to interface to currently available framegrabbers and signal processor boards.

3. Operation

The SICam™ performs the following operations which may be downloaded from the HOST or macro-programmed in the CCM software.

- Programmable CID Parameters:
  - imager reference voltages,
  - sub-array origin,
  - sub-array size,
  - number of NDRO read-out cycles,
  - video signal gain.
- CID Global Operations:
  - CID pixel well reset to light integrating mode,
  - global knockdown to clear imager charge above a programmed threshold,
  - global injection to clear all imager charge.
- CID Sub-array Operations: sub-array knockdown to clear charge in selected pixels above a programmed threshold,

- sub-array injection to clear all charge in selected pixels,
- non-destructive sub-array read-out.

- Camera Monitor & Control Operations:
  - external I/O for control & synchronization of external equipment,
  - HOST monitoring of SICam's critical signal levels and status,
  - temperature read-out of imager and FPA driver circuitry,
  - μP & LCA software downloadable from host computer.

4. Hardware Setup for Operation

To monitor the vacuum condition inside the dewar, a Varian thermocouple vacuum gauge was mounted onto the molecular sieve access cover. After all the O-ring surfaces of the mating flanges were carefully inspected and re-greased, the dewar was ready for evacuation .

4.1 Evacuation of Dewar

Follow the instructions (Appendix D) to evacuate the dewar to 5 militorr. If necessary, heat the dewar

with a blow-dryer to assist in evacuating the dewar. A leak was found after the molecular sieve access cover with the vacuum gauge was assembled. Since the dewar could not hold the liquid nitrogen for more than 6 hours, the vacuum deteriorated faster than expected. After the mating flanges were carefully re-greased and assembled, the dewar could hold the vacuum and liquid nitrogen for longer than 12 hours at 77K.

4.2 Cooling Down of the Dewar

The instruction manual suggests using a pressured liquid nitrogen fill line. There was no such device available at the time the test was performed. An alternate method of filling the liquid nitrogen was developed. The steps are:

- Orient the dewar so that it is vertical with the fill port facing up.
- Loosen the removable neck tube, insert the fill tube provided and leave the knurled knob loose.

- Connect a funnel to the fill tube port extending out of the fill port axially.
- Keep the funnel filled for about ten minutes, when the fill port is cooled down, remove the fill tube from the fill tube.
- Pour the liquid nitrogen slowly into the fill port until the liquid nitrogen tops off the fill port (about 20-30 minutes).

If the dewar is used vertically with fill port facing up, the removable neck tube should be left loose. If the dewar is used horizontally, turn the dewar towards the direction with the electronic box on the bottom and tighten the removable neck tube with the special tool provided immediately. It is the same as using the dewar vertically with the fill port facing down. After the dewar is full, it will take 30 minutes to one hour to allow the components to stabilize. The dewar will need to be refilled every 12 hours for longer hold time.

4.3 Setup of Interface and Cables

Every cable connects the electronics box to the dewar is unique except two coax cables with BNC connectors for the analog voltage signals, V_{EE} and V_{EO}. The 9-pin serial port on the camera control box is good for debugging the program through an ANSI terminal, but it slows the execution of the program in regular use. It should be left disconnected from the terminal when not in use. The set up of the interface board (National Instruments' AT-DIO-32F) is critical. The base I/O address (set by the on-board dip switch) of the board should be the same as what is defined in the header file for the "C" program. The DMA (direct memory access) channels used by the AT-DIO-32F board were found working only with DMA Channel 5 and 6. The working interrupt line setting is "none" for Group 1 and "IRQ11" for Group 2. The configuration is done by setting the jumpers on the board and saving a file by using the program called "DAQware" (daqconf.exe) provided by National Instruments.

5. Programming

Function libraries provided by CIDTEC and National Instruments enabled development of computer programs which control the SICam from a host computer. The function libraries were compiled with the Microsoft C compiler and currently available for the DOS environment only. The library functions communicate with the SICam through a National Instruments AT-DIO-32F board installed on the host PC's ISA bus. DMA transfers of image data are supported for storage in 640K base memory and extended memory.

During the test, some codes and parameters of the library functions as well as the hardware needed to be modified to function properly. All library functions return a non-zero error code if they fail. The program checks these returns and report errors in an applicable output format. More details on programming will be discussed in Appendix B.

III. Evaluation of the CID array

1. Introduction

In ground-based astronomical imaging, the limitation to the signal-to-noise ratio achievable at a particular irradiance level is determined by many parameters including the number of photons collected, blurring due to atmosphere, and detector non-uniformities. Techniques for building large-aperture telescopes have flourished in the last decade and techniques to remove pixel-to-pixel variations across the array have been developed. Considerable progress has also been made in compensating the incoming wavefront for phase variations introduced by the atmosphere so as to provide as sharp an image as possible. Most of the

image blur introduced by the atmosphere is first-order image motion, i.e. a translation of the air on the focal plane (Fried [1965]). The read-out architecture of a Charge Injection Device allows a sub-section of the array to be read while the remainder of the array continues to integrate. A sufficiently bright star located in such a sub-array could be used to provide a correction signal to a piezo electronically driven mirror and thus minimize blurring due to first-order image motion.

To evaluate the performance of a CID device in such an application, a program of laboratory testing of the imaging properties of such arrays was instigated. A dewar housing and electronics module commercially referred to as SICam® (described by van Gorden *et al* [1993]) was supplied by Charge Injection Device Technologies (CIDTEC®). A CID-38 chip with 512 x 512 pixels of pitch 28 μm was inserted in the dewar for these tests. The CID-38 incorporates a preamplifier-per-row architecture, with row and column addressing

controlled by shift registers. This chapter describes the methods used and the results obtained in the initial tests. All the measurements described in this thesis were made with the CID array cooled to 77K.

2. Laboratory Measurements

2.1 CID Stability

Of particular concern for long integration time exposures is the stability of the array. This can be evaluated by monitoring the DC level of read-out obtained without opening the shutter (i.e. in the dark). This is referred to as a bias frame. The stability of the bias was found to depend on the injection voltage (V_{inj}). Two different injection functions were used in the test programs. The first one was called the sub-array injection function which does pixel-by-pixel injection through the sub-array set by the coordinator parameters passed to the function. The second is the global injection function, the row and

column electrodes of all the pixels are biased to collapse the pixel-wells and "inject" the charge packet into the substrate collector. To find an optimal value for V_{inj}, the CID array was exposed to a high incident light level, a global inject was performed and the array was then read-out four times at each trial V_{inj} value. The change in the mean level of those four frames is plotted in figure 6 as a function of V_{inj} (Table 1 in Appendix B). The optimum value for the injection voltage which give the least bias drift was found to be 8.03 Volts.

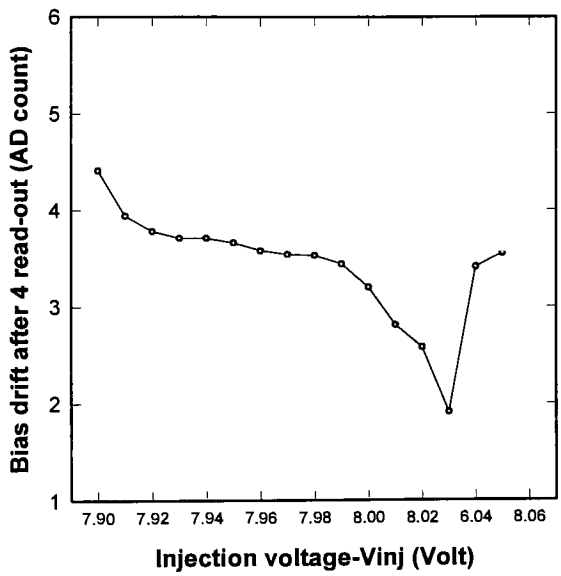


Figure 6 : Bias Drifting versus Injection Voltage.
(Using sub-array injection function)

To find an optimal value for V_{inj} a sequence of images were obtained while varying V_{inj}. The CID array was initially exposed to a high incident light level, a global inject of the CID was performed and the array was then read-out. This procedure was repeated at all the trial V_{inj} values. The change in the mean level between the before and after bias frames is plotted in figure 7 versus the V_{inj} voltage used (Table 2 in Appendix B). The optimum value (i.e. where the bias level changed least) for the injection voltage was found to be 7.95 Volts.

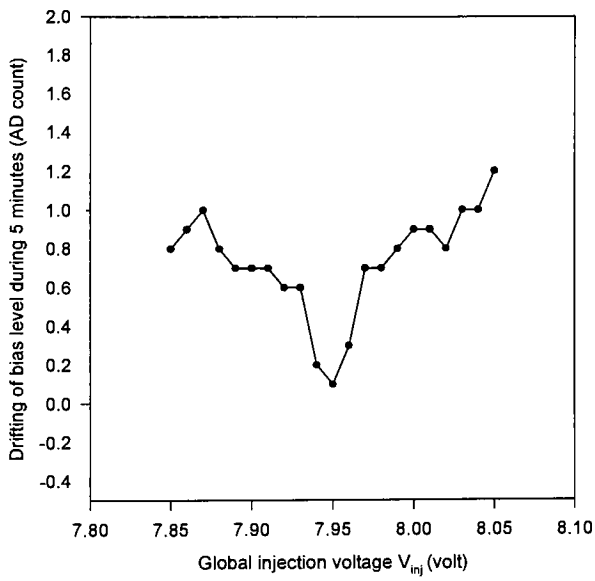


Figure 7 : Bias Drifting versus Injection Voltage.
(Using global injection function)

2.2 Read Noise

To determine the read noise and gain of the CID, the "photon transfer" technique[4] was used. The camera can be described in terms of five transfer functions, three of that are related to the sensor and the other two are related to the external signal processing circuitry. The input to the camera is given in units of incident photons, and the output of the camera is accomplished by encoding each pixel's signal into a digital number (DN). To convert the output signal S (DN) into fundamental physical units, it is necessary to find the appropriate factors (constants) to convert DN units into signal electrons. The constants can be found graphically by plotting a curve ("photon-transfer curve") of noise σ^2 (DN) as a function of signal S (DN). At low light level, the noise is dominated by the CID on-chip amplifier which is independent of the signal level. As the signal is increased, the noise eventually becomes dominated by the shot noise of the signal and is characterized by a line of slope 1/2.

The measured noise σ^2 after digitization in digital counts, or DN, is related to the signal level S (in DN), the read noise floor of N_r^2 (in electrons) and the conversion factor g (in DN/electron) by

$$\sigma^2 = (\sqrt{gS})^2 + g^2 N_r^2 \qquad (1)$$

In order to determine the values of g and N_r the CID was uniformly illuminated with a stabilized deuterium lamp. Exposure times were controlled by the shutter, in the range 1 to 50 seconds. The signal level in each of three randomly selected pixels on the array was read out 16 times at each exposure time (i.e. signal level). A plot of the mean signal level versus the variances of those 16 read-outs at each signal level is shown in figure 8. A least-squares fit to the equation gave a read noise N_r of 261 electrons with the gain of the system as $G=1/g=32.68e^-/DN$. These measurements were made using a read-out rate of 114 μsec per pixel (frequency of 8.8 kHz). This rate is set using the load gain function to pass the parameter to the SICam

electronics and referred to as "gain" (for the above measurements it was set to 255).

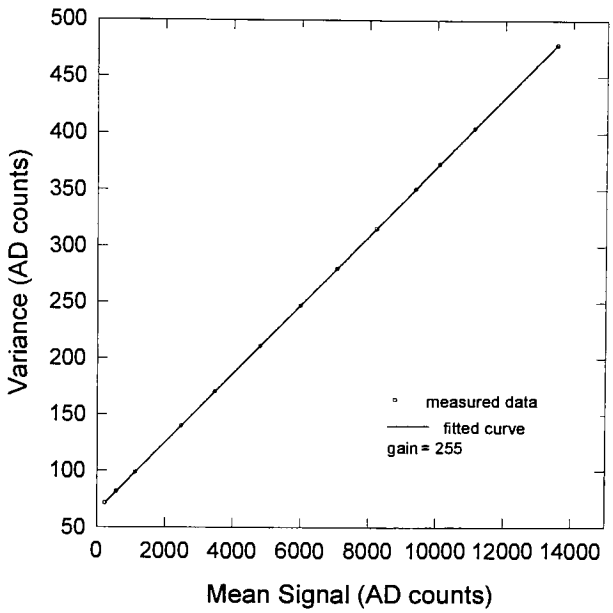


Figure 8 : Mean Signal Level versus Variance (gain = 255).

An alternate technique for determining the read noise was to read out the entire CID twice, again at a rate of 114 μsec/pixel (gain of 255) with the shutter closed. To eliminate the pattern noise due to the non-uniformity of the CID chip, these two read-outs were then subtracted and a histogram of the result is shown in figure 9. A gaussian distribution was fitted

to this histogram and a σ of 10.98 DN was found. By subtracting two bias frames, the noise was introduced by a factor of $\sqrt{2}$. This figure implies a read noise of $10.98 \times 32.68/\sqrt{2}$ electrons, or 254 electrons, similar to the result obtained using the photon transfer method.

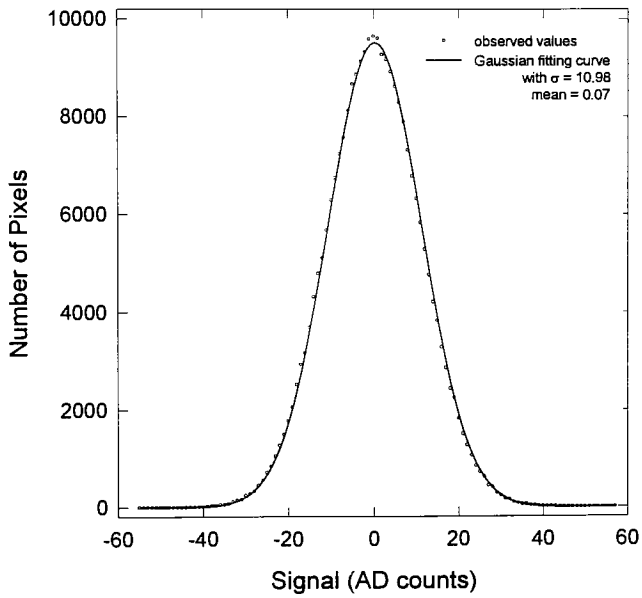


Figure 9 : Histogram of the Result of the Subtraction of Two Bias Frames.

2.3 Dark Current

Charge that accumulates without exposure to incoming photons is referred to as dark current. To evaluate the dark current for the CID38 at 77K, (three

hour integration of charge) were made with the shutter closed. Figure 10 shows a histogram of one such three-hour exposure after subtraction of a bias frame which was taken just prior to the beginning of the integration. Also shown is the gaussian distribution obtained from fitting to the histogram of the difference of two biases (as shown in figure 9). While there are some pixels with high dark current overall, the dark current for the majority of the array is not measurable.

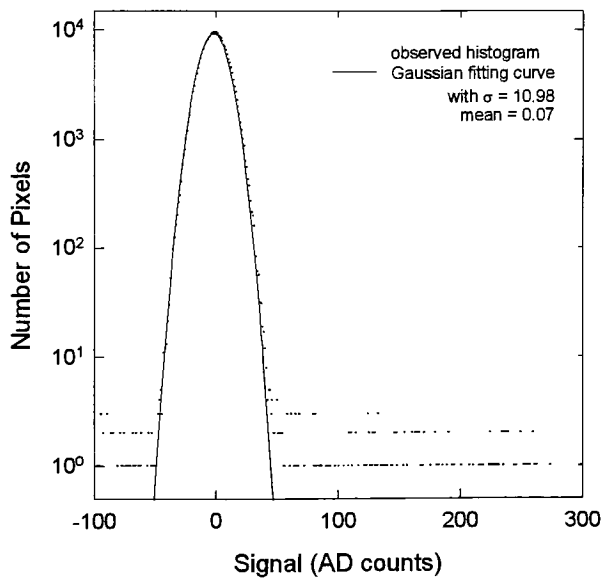


Figure 10: Histogram of bias subtracted 3 hour dark frame. Also shown is a Gaussian distribution with a σ equal to that found in figure 9.

2.4 Non-Destructive Read-out (NDRO)

During integration the column potential is greater than that of the row. At read-out the column potential is collapsed and charge transfers to the row potential. The charge is then sensed and may either be injected or the column potential reestablished and the charge transferred back to the column. Integration can now continue or the charge can be transferred again to the row pad and the signal measured again. This pixel operation is shown in figure 11. Such multiple sampling of the signal will allow for averaging of the temporal random noise.

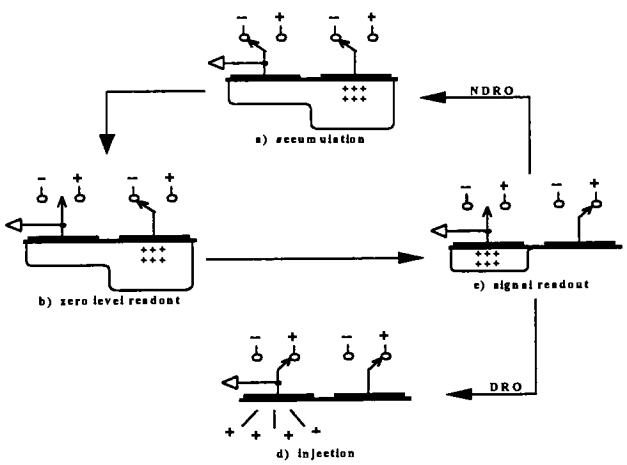


Figure 11: Pixel operation of non-destructive read-out (NDRO)

In order to assess the level of noise reduction possible using such non-destructive read-outs, the CID was globally injected and then a randomly selected pixel was read out non destructively 100 times after no exposure to light,. This procedure of injection and NDRO was repeated 100 times (Figure 12). A standard deviation was determined for the 100 values corresponding to the first read-out after each new injection. The average for the first four read-outs after each injection was then determined and the standard deviation of those 100 values was also determined. This continued until the average of the 100 read-outs following an inject was found and the standard deviation of these 100 values determined. These standard deviations were converted from AD counts to read noise in electrons using the earlier calibration and these value are plotted in figure 13. Also plotted in the figure is the reduction in read noise expected if the noise followed a gaussian noise form. The vertical location of the curve is arranged so

that it is the same as the noise measured for N=1. The noise does fall off with an increasing number of NDRO, in the form expected.

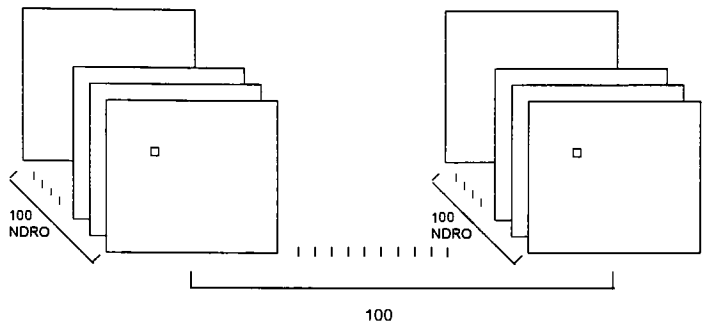


Figure 12 : Procedure of NDRO Noise Measurement.

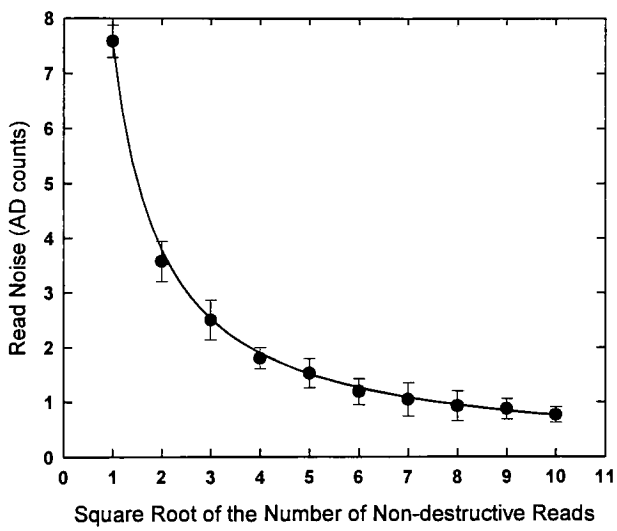


Figure 13 : Noise versus number of Non-Destructive Read-outs.

IV. Results

1. CID Stability

From figure 6 and figure 7, there are best values of injection voltages for avoid residue image. But while these values increase, the bias drift start to get worse. This is because the drive force from the storage pad drive the holes from the thin P-doped layer into the N-doped substrate. When this force reaches the boundary of the P-N junction, charges (holes) are completely "cleared" from being sensed. When the drive force gets cross the boundary of the P-N junction, the charges start to come back to cause the residue image.

There was no proper working global injection function code available at the beginning of the series of tests preferred for this thesis work. The sub-array

injection function was used to perform the global injection. It is a function which does pixel-by-pixel injection according to the sub-array parameters passed to it. The entire array was set to be the "sub-array", so the function worked as global injection equivalently.

After the macro code for the global injection function had been modified, it was found through the experiment that the stability of the bias was not only dependent on the value of the injection voltage applied, but also on the duration of the injection pulse, which is the parameter passed to the function. A residue image was found when using injection pulse with short duration (i.e. 100 msec). The injection voltage was set at the optimal value obtained from previous test. To find an optimal duration for the injection pulse, a bias frame was obtained both before and after the frame was exposed to a moderate light level. Figure 14 shows a plot of the injection pulse duration versus the difference in the mean level of the bias frame

injection of charges accumulated during the exposure to light. At injection, a voltage pulse of 7.95v is applied to the storage pad to build a eletronic field which moves the holes into the epitaxial substrate. Due to the collision, it needs a considerable long time of 800 msec (compare to 2 msec frome calculation assuming no collision) to move the holes into the substrate.

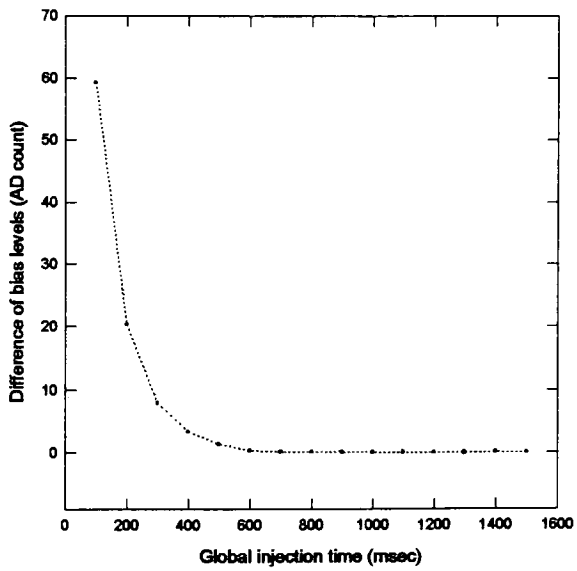


Figure 14 : Amount of bias drift versus injection pulse duration. (Using global injection function)

From the plots (Figure 6 and Figure 7), it was found that the bias is more stable at any injection voltage setting when using the global injection function

than using the sub-array injection function. One and half (1500 msec.) of injection pulse duration was used for the rest tests.

2. Read Noise

The curve fitting was down by using SigmaPlot®. A linear equation ($Y = aX + b$) is used as a fitted curve where X is the mean signal level S and Y is variance σ^2 (Table 4 in Appendix A). SigmaPlot® gives the values of **a** and **b**. From the equation (1) $\sigma^2 = \left(\sqrt{gS}\right)^2 + g^2 N_r^2$, we get $a=g$ and $b=g^2N_r^2$. From the table, the average gain $a=g=0.0306$ DN/electron or $G=1/g=32.68$ electrons/DN. The read noise $N_r=\sqrt{b}/a=261$ electrons.

The histogram of the result of the subtraction of the two bias frames was fitted to a gaussian distribution given by:

$$y = \frac{N}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (2)$$

where N is the total number of pixels on the CID38 chip (512x512), μ is the mean which equals the average pixel

value of the whole CID array, and σ is the standard deviation of all pixels. The figure shows that the histogram falls on the gaussian distribution curve perfectly. It indicates that the read noise of the CID is a gaussian distribution which is also true for CCDs[2]. The data is stored on the attached diskette since they are too many to list here. They are in SigmaPlot® for Windows file format.

3. Dark Current

The dark signal varies enormously among different pixels on the same CCD chip[2] and is a strong function of temperature. From the measurement (Figure 10), the number of pixels with high dark current is extremely low on the CID38 chip compared to the CCD[2] . This result is the benefit of the low working temperature. The CID chip works at liquid nitrogen temperature (-196°C) which is much lower than that (-35°C- -60°C) used for some high performance CCD chip.

4. Non-Destructive Read-out (NDRO)

As shown in Section 2.4 of Chapter III, the NDRO feature of the CID chip does decrease the noise by the factor of the square root of the number of read-outs. The data is shown in Table 5 in Appendix A.

There is an alternate way to prove that NDRO reduces the noise visually by averaging 36 and 100 frames of NDRO images. An average of 100 frames of bias was also taken. Subtract the average bias frame from the image of single read and each of average frame of 36 and 100 images. One hundred pixels of a row in a relative dark section of each frame was picked for the calculation. The values of these pixels are equivalent to reading the same pixel once, 36, and 100 times as described in Section 2.4 of Chapter III. The standard deviations for these three sets of data are 7.87, 1.63 and 1.18 (DN). Since the average of 100 frames of bias was subtracted out, the noises for the average imges of 36 NDRO and 100 NDRO are $\sigma_{100} = 1.18/\sqrt{2} = 0.834$, $\sigma_{36} = \sqrt{1.63^2 - 0.834^2} = 1.400$. For 36 and 100 NDRO image, the noise is reduced by a factor of 6 and 10 respectively.

The ratio between these two is 0.6. The ratio between 0.834 and 1.400 is 0.596. So it does fall on the NDRO theoretical curve. Figure 15 and 16 show the differences of noise comparing single read-out versus 36 NDRO and 100 NDRO at gain of 255.

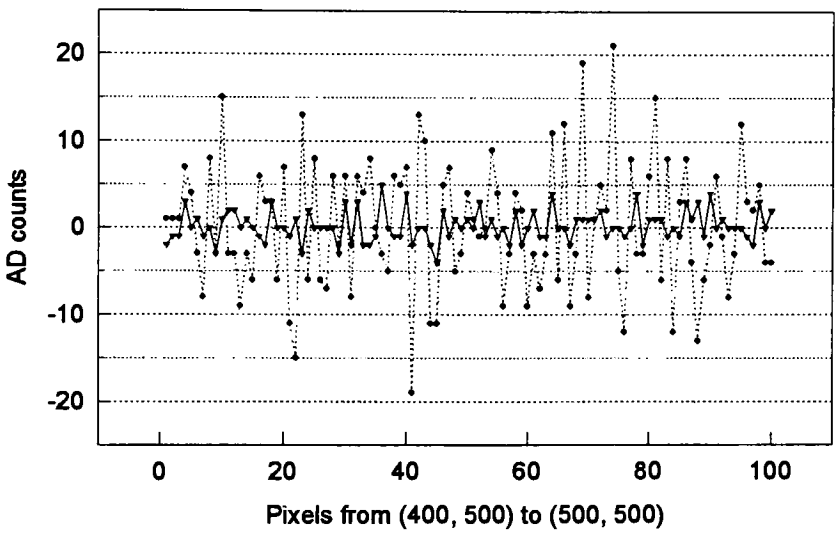


Figure 15 : Noise of single read-out versus 36 NDRO.

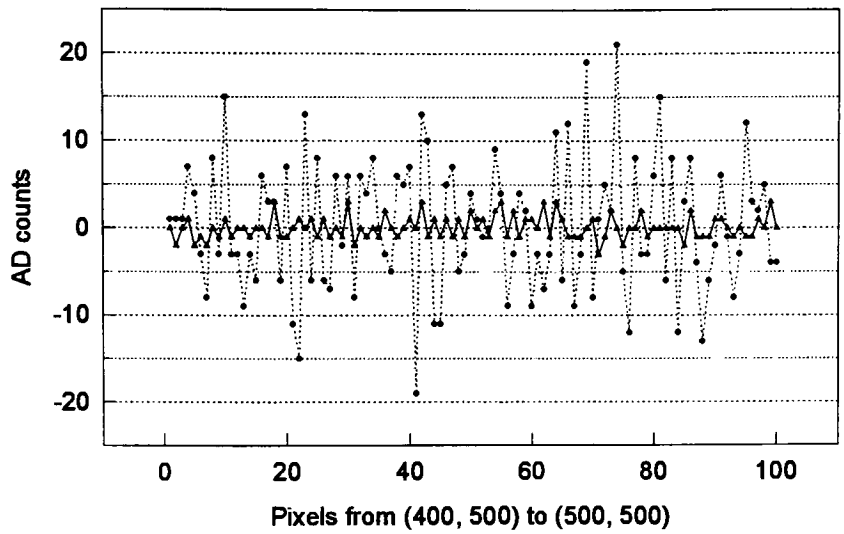


Figure 16 : Noise of single read-out versus 100 NDRO.

The following figures show the images of 100 NDRO comparing with image of single read-out at different gain. The images on the right have better quality.

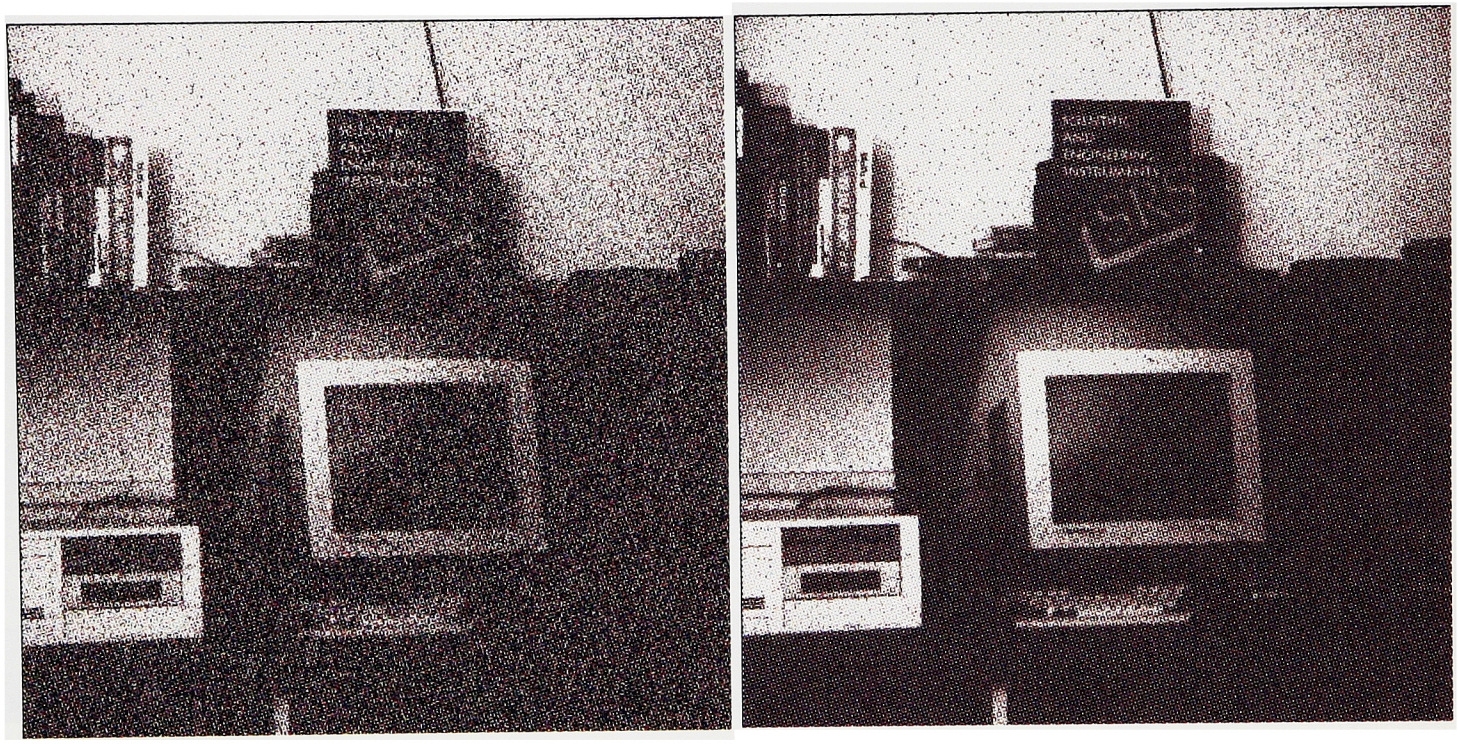


Figure 17 :Images of single read-out versus 36 NDRO at gain=255.
(injection voltage = 7.95, injection pulse duration = 1500msec.)

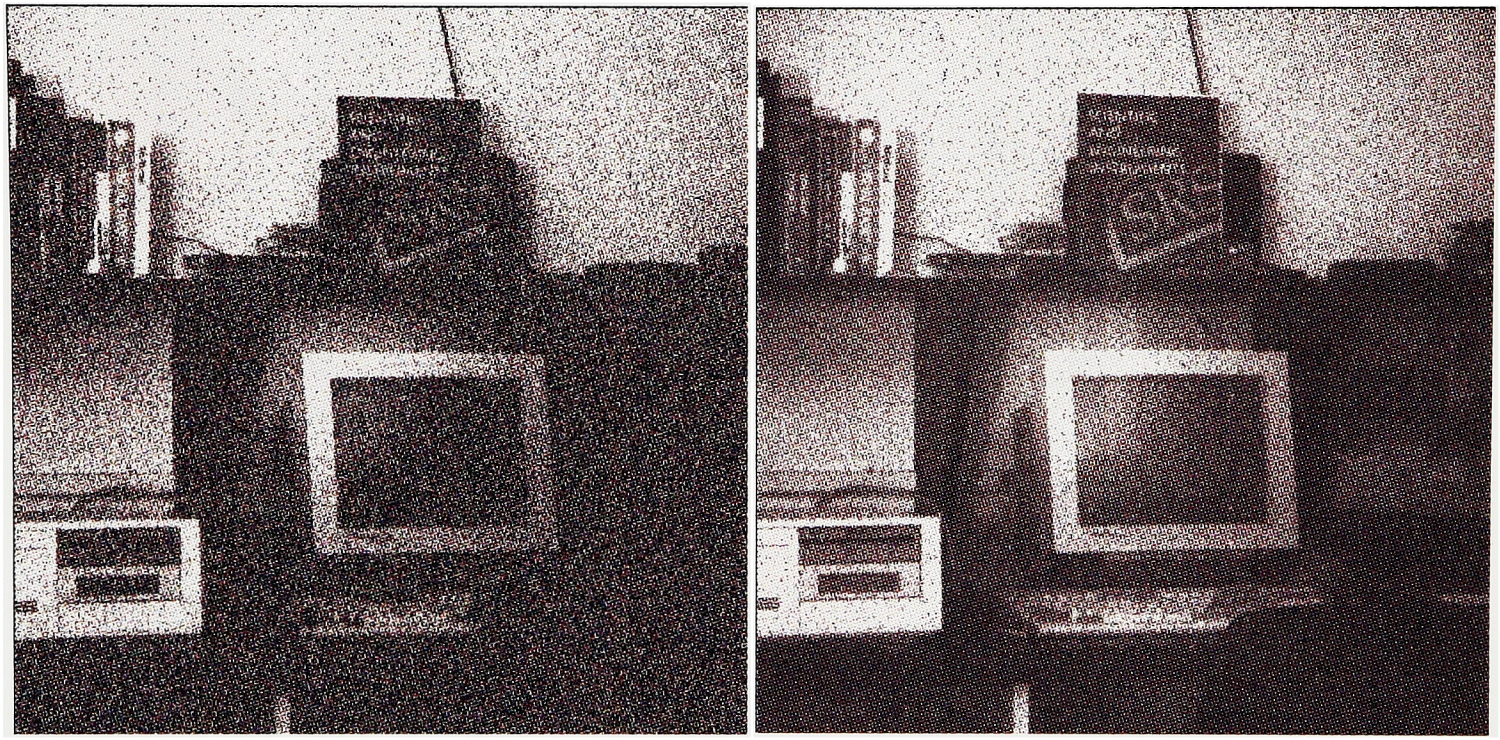


Figure 18 : Images of single read-out versus 100 NDRO at gain=255.
(injection voltage = 7.95, injection pulse duration = 1500msec.)

V. Conclusions/Recommendations

1. We have established a test procedure that can be used for evaluating the CID camera system for low light imaging application.

2. It is found that CID array is very stable for long integration time exposures with proper injection voltage and pulse duration.

3. CID array has much lower dark current comparing with CCDs for the same application.

4. CID's read noise is much higher than that of CCDs', but it can be significantly improved by using the CID's unique feature - nondestructive read-out.

5. The system has been tested practically and it is found that the CID camera system has the capability for the low light astronomical imaging application. Figure 16 and 17 show the images of star clusters taken through the 61-cm Cassegrain reflecting telescope at the University of Rochester's C.E.K. Mees Observatory.

6. The disadvantages are that the CID array read-out speed is slow due to the analog/digital conversion circuit design which is not upgradable. The electronics unit are relative larger compared to that for the CCD. The system has less portability.

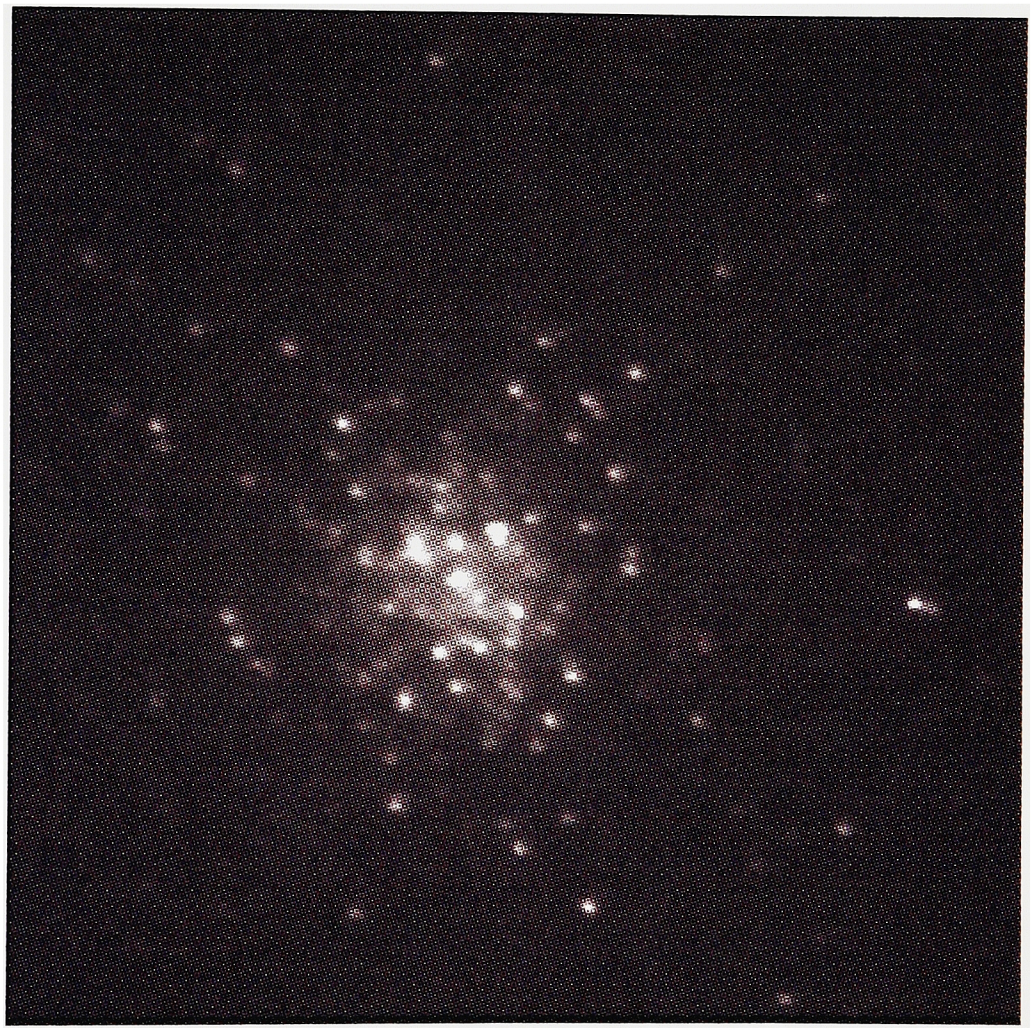


Figure 19 : CID image with a single read-out of star cluster M3.
(600 seconds at gain=255)

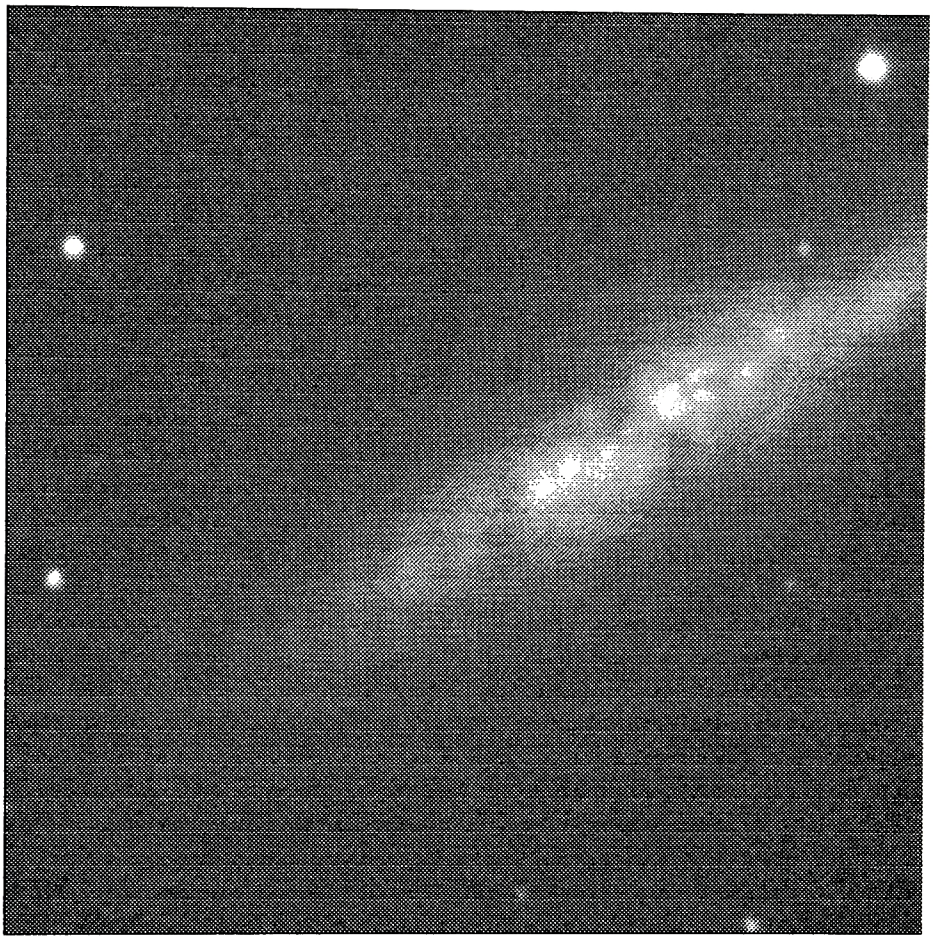


Figure 20 : CID image with average of 9 NDRO of M82.
(600 seconds at gain=255)

VI. Appendices

Appendix A. Tables

Table 1. Bias Drifting versus Injection Voltage (V_{inj}).
(Using sub-array injection function)
(Gain = 255, 1 AD = 32 e^-)

V_{inj} (volt)	7.90	7.91	7.92	7.93	7.94	7.95	7.96	7.97	7.98	7.99	8.00	8.01	8.02	8.03	8.04	8.05
drift (AD)	4.41	3.94	3.78	3.71	3.71	3.66	3.58	3.54	3.53	3.44	3.20	2.81	2.58	1.91	3.41	3.55

Table 2. Bias Drifting versus Injection Voltage (V_{inj}).
(Using global injection function)
(Gain = 255, 1 AD = 32 e^-)

V_{inj} (volt)	7.85	7.86	7.87	7.88	7.89	7.90	7.91	7.92	7.93	7.94	7.95
drift (AD)	0.80	0.90	1.00	0.80	0.70	0.70	0.70	0.60	0.60	0.20	0.10

V_{inj} (volt)	7.96	7.97	7.98	7.99	8.00	8.01	8.02	8.03	8.04	8.05
drift (AD)	0.30	0.70	0.70	0.80	0.90	0.90	0.80	1.00	1.00	1.20

Table 3. Bias drift versus injection pulse duration.
(Using global injection function)
(Gain = 255, 1 AD = 32 e^-)

GI time (msec)	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500
Bias Drift (AD)	59.28	20.34	7.89	3.34	1.34	0.33	0.12	0.05	0.07	0.06	0.06	0.04	0.06	0.05	0.04

Table 4. Read Noise Measurement

Pixel (100,100)		Pixel (255,255)		Pixel (400,400)	
Mean Signal-S	Variance-σ^2	Mean Signal-S	Variance-σ^2	Mean Signal-S	Variance-σ^2
222	70.7281	254	71.7409	270	72.2500
488	78.8544	579	81.7216	578	81.7216
982	94.0900	1136	98.8036	1148	99.2016
2029	126.1100	2479	139.9489	2460	139.2400
2878	152.0300	3476	170.5636	3416	168.4804
4057	188.2384	4814	211.4116	4746	209.0916
5029	217.8586	5994	247.4300	5966	246.4900
5978	246.8041	7068	280.2276	7084	280.8976
7023	278.8900	8226	315.7729	8189	314.7076
7932	306.6001	9387	351.1876	9310	348.9424
8925	337.0896	10088	372.8761	10100	372.8761
9766	362.9025	11122	404.4121	12835	456.6769
11943	429.3184	13541	478.2969	13549	478.7344
Curve fitting values		Curve fitting values		Curve fitting values	
a = 0.0306	b = 63.9910	a = 0.0306	b = 64.0548	a = 0.0306	b = 63.9914
Read Noise		Read Noise		Read Noise	
N_r = 261.4195		N_r = 261.5498		N_r = 261.4203	

Table 5. Non-destructive Read-out

(Gain = 255, 1 AD = 32 e^-)

$\sqrt{\#\text{ of NDRO}}$	1	2	3	4	5	6	7	8	9	10
Noise (DN)	7.5882	3.5763	2.5043	1.8014	1.5257	1.1914	1.0475	0.9274	0.8721	0.7681

Appendix B. SCM5000E Programs

Generally the control program will follow the work flow listed below:

- Initialize program variables,
- Initialize the NI AT-DIO-32F interface board,
- Initialize the control unit (SCM5000E) hardware,
- Download 80286 and/or LCA software if updating local SCM software. Otherwise, skip to next item and SCM EPROM software will be used,
- Download default parameters to the SCM,
- Verify operation of SCM by checking critical PS levels and HOST communications,
- Download calibration files to the SCM,
- Download image acquisition parameters to the SCM,
- Allocate image buffers in local 640K memory or extended memory as appropriate,
- Perform image acquiring operations,

- Display and/or Save image data into a file with certain format which can be used by other software as appropriate,
- Free allocated image buffers before terminating.

CID control program in Microsoft C:

```
/*****************************************************************
FUNCTIONS INCLUDED IN THIS MODULE:
      FUNCTION            DESCRIPTION

      initSCM()           initializes SCM

      calSCM()            perfomrs SCM calibration

      readSCM()           reads subarray data from SCM

      main()              mainline function controlling system
*****************************************************************
main(): main function for CID control program
*****************************************************************
DESCRIPTION:
- calls scm functions to initialize, calibrate and acquire data
*****************************************************************/
void main(void)
{
unsigned indx;           // index variable

      printf("\n\n\t\tSCM5000E Control Code -- %s\n", revMSG);

      printf("\nInitializing SCM:\n");
      exRSLT = initSCM();              // initialize SCM
      if(ckEXerr("initSCM"))  goto errEXIT;

      printf("\n\nCalibrating SCM:\n");
      exRSLT = calSCM();               // calibrate SCM
      if(ckEXerr("calSCM"))   goto errEXIT;
```

```
      printf("\n\nReading SCM Subarray: \n");
      for(indx=0; indx<300; indx++)
         {
         exRSLT = readSCM();      // read data from SCM subarray
         if(ckEXerr("readSCM"))  goto errEXIT;
         }

      printf("\n\nControl Program Complete\n");
      exit(0);

errEXIT:                                // close shutter
      scmRSLT = scm_CTRLshutter(3, scmWAIT);
      exit(-1);
}
/* INCLUDE FILES */
#include <process.h>
#include <stdlib.h>
#include <conio.h>
#include <stdio.h>
#include <ctype.h>
#include <malloc.h>
#include <graph.h>
#include "example.h"                        // Example header file
#include "..\libs\scm_dos.h"                // SCM header file

/* GLOBAL VARIABLE DECLARATIONS */
int     exRSLT;                 // result of function call
calDATA offcDATA[1];            // data for OFFC cal
calDATA offpDATA[512];          // data for OFFP cal
calDATA offrDATA[2];            // data for OFFR cal

/* ASCII STRING DECLARATIONS */
char    revMSG[] = {"V0.01 02FEB94"};

/* FUNCTIONS */
/*****************************************************************
initSCM(): initializes the SCM after power-on or reset
*****************************************************************
DESCRIPTION:
      - displays error analysis screen if error occured
RETURN PARAMETER:
        if no error:  scmOK.......0
   —    if error:     scmERROR....1
*****************************************************************/
int   initSCM(void)
{
      scmRSLT = scm_CLReLOG();        // clear SCM error
```

```
        if(scm_CKerror(CLReLOG)) return(scmERROR);

        scmRSLT = scm_INITcomm();        // initialize SCM communications
        if(scm_CKerror(INITcomm)) return(scmERROR);

        scmRSLT = scm_RSTscm();          // reset SCM
        if(scm_CKerror(RSTscm)) return(scmERROR);

        // load RAM based 286 code
        // scm286.bin must be in same directory

        scmRSLT = scm_LD286PRG("SCM286.bin", 0x8000, 0x1000, 0x0000,
                               0x1000, 0xF000);
        if(scm_CKerror(LD286PRG)) return(scmERROR);
        printf("\t\t...RAM based 286-Code loaded\n");

        scmRSLT = scm_LDlcaPRG("scmlca.bin");
        if(scm_CKerror(LDlcaPRG)) return(scmERROR);
        printf("\t\t...EPROM based lca-Code loaded\n");

        // initialize SCM default variables
        scmDFLT.actCOL = 512;            // override default row settings
        scmDFLT.actROW = 512;            // override default row settings
        scmDFLT.offE   = (float)0.0;
        scmDFLT.offO   = (float)0.0;
        scmDFLT.offP   = (float)0.0;
        scmDFLT.offC   = (float)0.0;

        scmRSLT = scm_LDdefPARAMS(scmDFLT);
        if(scm_CKerror(LDdefPARAMS)) return(scmERROR);
        printf("\t\t...Defaults Loaded\n");
        return(scmOK);
}

/*****************************************************************
calSCM(): performs calibration on SCM & downloads data to SCM
*****************************************************************
DESCRIPTION:
        performs OFFP & OFFR calibrations
        downloads OFFP & OFFR data to SCM
RETURN PARAMETER:
        if no error:  scmOK.......0
      - if error:     scmERROR:...1
*****************************************************************/
int     calSCM(void)
{
unsigned Pstat;                  // load PASS/FAIL status
```

```
// close shutter
scmRSLT = scm_CTRLshutter(3, scmWAIT);
if(scm_CKerror(CTRLshutter)) return(scmERROR);

// calibrate OFFC
printf("\t\t...calibrating OFFC\n");
scmRSLT = scm_GEToffcCAL(150, (float)0.25, offcDATA,
                         scmDFLT.actCOL, scmDFLT.actROW);
if(scm_CKerror(GEToffcCAL)) return(scmERROR);

// load OFFC
printf("\t\t...loading OFFC\n");
scmRSLT = scm_LDoffcCAL(offcDATA, &Pstat);
if(scm_CKerror(LDoffcCAL)) return(scmERROR);

if(!Pstat)
printf("\t\t\t\t...SCM OFFC Calibration Failed\n");

// calibrate OFFP
printf("\t\t...calibrating OFFP\n");
scmRSLT = scm_GEToffpCAL(150, (float)0.25, offcDATA, offpDATA,
                         scmDFLT.actCOL, scmDFLT.actROW);
if(scm_CKerror(GEToffpCAL)) return(scmERROR);

// load OFFP
printf("\t\t...loading OFFP\n");
scmRSLT = scm_LDoffpCAL(offpDATA, scmDFLT.actROW, &Pstat);
if(scm_CKerror(LDoffpCAL)) return(scmERROR);
if(!Pstat)
printf("\t\t\t\t...SCM OFFP Calibration Failed\n");

// calibrate OFFR
printf("\t\t...calibrating OFFR\n");
scmRSLT = scm_GEToffrCAL(150, (float)0.25, (float)0.5, offcDATA,
                        offrDATA, scmDFLT.actCOL, scmDFLT.actROW);
if(scm_CKerror(GEToffrCAL)) return(scmERROR);

// load OFFR
printf("\t\t...loading OFFR\n");
scmRSLT = scm_LDoffrCAL(offrDATA, &Pstat);
if(scm_CKerror(LDoffrCAL)) return(scmERROR);

if(!Pstat)
printf("\t\t\t\t...SCM OFFR Calibration Failed\n");

printf("\t\t...Calibration Complete\n");
return(scmOK);
```

```
}

/*****************************************************************
readSCM(): read SCM subarray data from SCM
*****************************************************************
DESCRIPTION:
      - loads gain, number of reads, subarray start & size
        reads subarray data
RETURN PARAMETER:
        if no error:  scmOK.......0
        if error:     scmERROR....1
*****************************************************************/
#define RGB(r,g,b) (((long) ((b) << 8 | (g)) << 8) | (r))

long tmp, pal[256];

int     readSCM(void)
{
#define horz    512       // subarray width
#define vert    512       // subarray height
unsigned far *rBUF, *rBUF_swab;// pointer to read memory allocated below
unsigned    i, j, k, l;         // index variable
FILE    *fptr;
unsigned long   xIDX;
int *XMShandleptr;
unsigned int gain, read_count, nBUF, frame_count;
unsigned long shutter;
unsigned red, blue, green;
char chara, header[80], header1, kbchar;
char filename[80], fitsfile[20];
unsigned int data, *swab_data;
float Vdata;
unsigned Bdata;

      for(i=0; i<256; i++){     // convert 256 colors to 256 gray levels
         red = i;
         blue = i;
         green = i;
         pal[i] = RGB(red, green, blue);
      }

      printf("\nDefault settings:\
              \n                  gain =255, \
              \n                  read =  1, \
              \n          exposure time =  0 msec, \
              \n            frame count =  1, \
              \nUsing default settings?(y/n) ");
```

```
if((kbchar= (char)getch()) == 'n'){
   printf("\nEnter gain (0-255):");
   scanf("%u", &gain);
   printf("\nEnter number of read per pixel: ");
   scanf("%u", &read_count);
   printf("\nEnter exposure time (msec):");
   scanf("%lu", &shutter);
   printf("\nEnter number of frames (up to 30) to be taken: ");
   scanf("%u", &frame_count);
   printf("\nEnter FITS image file basename (5 letters): ");
   scanf("%s", fitsfile);
}
else if(kbchar == 'y'){
   gain = 255U;
   read_count = 1U;
   shutter = 0L;
   frame_count = 1U;
}


scmRSLT = scm_LDgain(gain);       // load gain
if(scm_CKerror(LDgain)) return(scmERROR);

// load number of reads per PIX
scmRSLT = scm_LDnumREADS(read_count);
if(scm_CKerror(LDnumREADS)) return(scmERROR);

// load subarray start address
scmRSLT = scm_LDsubSTART(0, 0);
if(scm_CKerror(LDsubSTART)) return(scmERROR);

// load subarray size
scmRSLT = scm_LDsubSIZE(512, 512);
if(scm_CKerror(LDsubSIZE)) return(scmERROR);

// read CID array temperature
scmRSLT = scm_SETtempPWR(1);
scmRSLT = scm_RDauxAD(0, &Bdata);
Vdata = scm_CONVaadB2V(0, Bdata);
scmRSLT = scm_SETtempPWR(0);
printf("\nThe temprature is %7.2f C.\n", (Vdata+91.));

if((rBUF_swab=(unsigned far *) halloc(16128*sizeof(int),
                sizeof(int))) == NULL){
   printf("\t\t\t\t...Swab byte Memory Allocation Failed\n");
   _hfree(rBUF_swab);           // free memory possibly allocated
}
```

```
if((rBUF=(unsigned far *) halloc(16128*sizeof(int), sizeof(int)))
          == NULL){
   printf("\t\t\t\t...Read Memory Allocation Failed\n");
   _hfree(rBUF);                  // free memory possibly allocated
   return(scmERROR);
}
else{
   for(k=0; k<frame_count; k++){
      if(k == 0){
         printf("\nOpen shutter?(y/n)");
         kbchar = (char) getch();
         printf("%c", kbchar);
         scmRSLT = scm_GLBinject(700);  // clear CID pixel charge
         if(scm_CKerror(GLBinject)) return(scmERROR);

         if(kbchar == 'y'){
            scmRSLT = scm_CTRLshutter(0, scmWAIT);
            if(scm_CKerror(CTRLshutter)) return(scmERROR);
         }
         scm_WAITmsec(shutter);           // wait for exposure
                                          // close shutter
         scmRSLT = scm_CTRLshutter(3, scmWAIT);
         if(scm_CKerror(CTRLshutter)) return(scmERROR);

      }
      printf("\n Taking frame %u...", k+1);
      scmRSLT = scm_XMSmemALLOC((int far *)&XMShandleptr[1],
                                (long)horz*(long)vert);
      if(scm_CKerror(XMSmemALLOC)) return(scmERROR);
      scmRSLT = scm_SUBreadXMS(XMShandleptr[1], scmNORMAL);
      if(scm_CKerror(SUBreadXMS)){
         scmRSLT = scm_XMSmemFREE(XMShandleptr[1]);
         return(scmERROR);
      }
      if(frame_count > 200){
         scmRSLT = scm_XMSmemFREE(XMShandleptr[1]);
         return(scmERROR);
      }

      sprintf(filename, "%s%d.fit", fitsfile, k);
      // open a binary file to write data in FITS format

      if((fptr = fopen(filename, "wb")) == NULL){
         printf("File could not be opened\n");
         return(scmERROR);
      }
```

```
        // write FITS header
        sprintf(header, "%-80s", "SIMPLE  =                    T");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "BITPIX  =                   16");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "NAXIS   =                    3");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "NAXIS1  =                  512");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "NAXIS2  =                  512");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "NAXIS3  =                    1");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "COMMENT =                     ");
        fwrite(header, 1, 80, fptr);
        sprintf(header, "%-80s", "END                           ");
        fwrite(header, 1, 80, fptr);
        for(j=0; j<28; j++){
           sprintf(header, "%-80s", " ");
           fwrite(header, 1, 80, fptr);
        }

        for(nBUF=0, xIDX=0; nBUF<16; nBUF++, xIDX+=16128L){
           scmRSLT = scm_XMSmemCOPY(XMShandleptr[1], rBUF, xIDX,
                                    16128L, scmXMS2BASE);
           if(scm_CKerror(XMSmemCOPY)) return(scmERROR);
           _swab((char *)rBUF, (char *)rBUF_swab, 32256);
           fwrite(rBUF_swab, sizeof(int), 16128, fptr);
        }

        scmRSLT = scm_XMSmemCOPY(XMShandleptr[1], rBUF, xIDX, 4096,
                                 scmXMS2BASE);
        if(scm_CKerror(XMSmemCOPY)) return(scmERROR);
        _swab((char *)rBUF, (char *)rBUF_swab, 8192);  //max 32767
        fwrite(rBUF_swab, sizeof(int), 4096, fptr);
        fclose(fptr);
    }
}

scmRSLT = scm_XMSmemFREE(XMShandleptr[1]);
if(scm_CKerror(XMSmemFREE)) return(scmERROR);

_hfree(rBUF);              // free memory for read
_hfree(rBUF_swab);

k = 0;
printf("\n Would you like to look at any image file?(y/n) \n");
```

```
      if((chara = getch()) == 'n') goto quit;

      do{
         printf("\nEnter FITS image file name (********.fit): ");
         scanf("%s", fitsfile);
         if((fptr = fopen(fitsfile, "rb")) ==NULL){
            printf("File could not be opened");
            exit(0);
         }              // open the FITS file for display

         for(j=0; j<36; j++){              // read the header off
            for(i=0; i<80; i++){
               fscanf(fptr, "%c", &header1);
            }
         }

         if(_setvideomode(_SRES256COLOR)){
            _remapallpalette( pal );        // redefine the palette
            _rectangle(_GBORDER, 142,42,658,558);
            for(i=0; i<512; i++){
               for(j=0; j<512; j++){
                  fread(&data, 2, 1, fptr);
                  _swab((char *)&data, (char *)swab_data, 2);
                  //swab bytes back for display on monitor
                  _setcolor((short)(*swab_data/256));
                  _setpixel((short)(144+j),(short)(44+i));
               }
            }
            _getch();
            _setvideomode(_DEFAULTMODE);
         }
         fclose(fptr);
         printf("\nLook another image?(y/n) ");
         chara = (char )getch();
      }while(chara == 'y');


quit:
      printf("\n press any key to continue or \"q\" to quit \n");
      if((chara = getch()) == 'q') exit(0);

      return(scmOK);
}  —
```

Appendix C. Specifications of SICam (SCM5000e)

CID38 Sensor (512H x 512V)

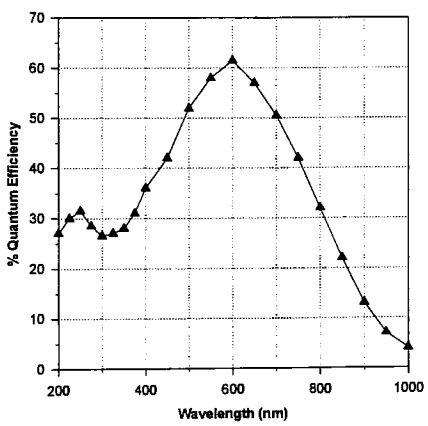
Sensor format:

Active Pixel Elements	512H x 512V
Active Sensor Area	14.336 x 14.336mm
Pixel Size	28.0 x 28.0μm
Image Format	20.274mm diagonal
Geometric Accuracy	Better than 0.001%

Sensor Characteristics:

Full Well Capacity	1.2 Me/pixel@5.25v
Dark Current (LN_2)	<0.01 e/pixel/sec
Sensitivity	200 to 700 photo electrons
ADC Dynamic Range	14 bits w/o Non-destructive Readout (NDRO);
Response Uniformity	Better than 5%
Read Noise	200 electrons rms(single read) 20 electrons rms(after 100 NDRO)
Linear Dynamic Range	> 8 orders of magnitude w/100 sec integration

Spectral Response:



Appendix D. Instructions for System Setup

Initial Setup

Before each cool down of the dewar, the following few items should be check to insure proper performance of the system.

1. Replace the molecular sieve with regenerated molecular sieve if the dewar is vented to atmosphere for longer than one hour.

2. Inspect and re-grease if necessary all the O-ring surfaces of the mating flanges that are taken apart by the user to access the dewar's internal components.
3. Insure that O-rings are keyed into flanges to avoid pinching and damaging the O-rings when assembling mating flanges.

4. Avoid contact with the dewar's inner surface and components with bare skin. This leaves oil residues that are difficult for the vacuum pump to remove and will cause unnecessary out-gassing during operation.

Evacuating the Dewar

The following steps will assists in evacuating the dewar for use.

1. Assemble all components and dewar vacuum housing and check that all the O-rings of the disassembled parts are greased and undamaged.

2. Make sure that the snap ring that holds the actuator plug for shipping is removed.

3. Connect the operator to the actuator valve and tighten the hex nut. The operator will feel loose on the actuator even after the nut is tightened.

4. Push in and turn the operator knob to engage the operator in the actuator plug.

5. The valve is then opened by retracting the operator knob and pulling out the plug.

6. After the dewar is evacuated for use, push in the operator to insert the plug and unscrew the knob so the operator can be removed.

7. Loosen the hex screw and remove the operator.

Cool Down of the Dewar

Once the dewar has been sufficiently evacuated, it may be cooled to liquid nitrogen temperature.

1. Orientate the dewar so that it is either vertical or laying flat with the fill port facing up.

2. Insert the fill tube provided and tighten the knurled knob.

3. Connect a liquid nitrogen fill line to the fill tube port extending out of the fill port exially. The port extending out 90 degrees is for venting and should be directed away from people and items sensitive to liquid nitrogen temperatures.

4. With the initial fill, the liquid nitrogen may be expelled at a quick rate because of the warm vessel and may appear to be full. However, you should continue to

fill the dewar until a steady stream of liquid is released.

5. After the dewar is full, it will take a few minutes to allow the components to stabilize. This event will be marked by a quick rush of gas exiting the dewar. Since a significant amount of liquid nitrogen was boiled-off in cooling the dewar, the dewar will need to be re-filled before use to give the required hold time.

VII. References

1. Z. Ninkov, C. Tang, R. L. Easton, "*Evaluation of a Charge Injection Device Array*", SPIE Proc. vol. 2172, pp. 15, [1994]

2. Z. Ninkov, B. Backer, D. Bretz and P. Burns, "Characterization of a Large Format CCD Array", SPIE Proc. vol. 1987, pp. 14-27, [1993]

3. S. VanGorden, J. Hutton, D. Fassett, J. Carbone, "*SICam - A PC Compatible Digital Instrumentation Camera Incorporating a Large Pixel 512x512 format Charge Injection Device (CID)*", SPIE Proc. vol. 1901, pp. 83, [1993]

4. J. Zarnowski, E. Eid, F. Arnold, M. Pace, J. Carbone and B. Williams, "*Performance of a Large Format Charge Injection Device*", SPIE Proc. vol. 1900, pp. 110, [1993]

5. El-Sayed Eid, *"Design of a Large Format Charge Injection Device Imager for Spectroscopy"*, High Resolution Sensors and Hybrid Systems Conference, Feb. 1992

6. A. M. Fowler and I. Gatley, "*Noise Reduction Strategy for Hybrid IR focal plane arrays*", *SPIE* Proc. vol. *1541*, pp. 127, [1991]

7. J. Carbone, "*Application of Low Noise CID Imagers in Scientific Instrumentation Cameras*", SPIE Proc. vol. 1447, pp. 229-242, [1991]

8. C. Buil, "*CCD Astronomy : Construction and Use of an Astronomical CCD camera*", Richmond, Va. : Willmann-Bell, c1991

9. R. Berry, "*Introduction to Astronomical Image Processing : A Comprehensive Guide to CCD Image Enhancement for the IBM-PC and Compatibles With ImagePro Software*", Richmond, Va. : Willmann-Bell, 1991

10. M. Pilon, Ph.D. Dissertation, University of Arizona, [1991]

11. M. Pilon, "*Evaluation of a New Array Detector Atomic Emission Spectrometer for Inductively Coupled Plasma Atomic Emission Spectroscopy*", Applied Spectroscopy, vol. 44, Nov. 10, 1990

12. J. Carbone, "*Sub-pixel Interpolation With Charge Injection Devices (CID's)*", SPIE vol. 1071, Optical Sensors and Electronic Photography, pp. 80-89, [1989]

13. G. Michon, "*CID Image Sensor With Parallel Reading of All Cells in Each Sensing Array Row*", United States Patent #4807-038, Feb. 21, 1989

14. "*Final Report on Radiation Testing of CID 73E Charge Injection Device*", General Electric Company, January 1988

15. J. R. Janesick, K. P. Klassen and T. Elliott, "*CCD Charge Collection Efficiency and the Photon Transfer Technique*", Optical Engineering 26 (10), 972, [1987]

16. "*Optoelectronics and Image-sensor Data Book : CCD Image Sensors, Optocouplers, Intelligent Displays, IR Emitters, and Phototransistors*", Dallas, Tex. : Texas Instruments, c1987

17. G. Michon, "*CID Image Sensor With a Preamplifier for Each Sensing Array Row*", United States Patent #4689-688, Aug. 25, 1987

18. J. Janesick, el al, "*CCD Charge Collection Efficiency and the Photo Transfer Technique*", UV Technology, R. Huffman, Ed., SPIE Proc. 687, PP. 36, [1986]

19. G. Michon, "*CID Image Sensor With Improved Sensitivity*", SPSE Conference on Electronic Imaging, Ict. 13-17, [1986]

20. D. Brown, M. Ghezzo, M. Garfinkel, "*Transparent Metal Oxide Electrode Imager*", IEEE Transactions on Electron Devices, ED-23(2), 196 [1976]

21. G. Michon, "*Method and Apparatus for Sensing Radiation and Providing Electrical Readout*", United States Patent #3,768,263, Jan. 15, 1974