IPN Progress Report 42-155 November 15, 2003

Design of an Event-Driven, Random-Access, Windowing CCD-Based Camera

S. P. Monacos,[1] R. K. Lam,[1] A. A. Portillo,[2] D. Q. Zhu,[3] G. G. Ortiz[2]

Commercially available cameras are not designed for a combination of single-frame and high-speed streaming digital video with real-time control of size and location of multiple regions-of-interest (ROIs). A message-passing paradigm is defined to achieve low-level camera control with high-level system operation. This functionality is achieved by asynchronously sending messages to the camera for event-driven operation, where an event is defined as image capture or pixel readout of a ROI, without knowledge of detailed in-camera timing. This methodology provides a random access, real-time, event-driven (RARE) camera for adaptive camera control and is well suited for target-tracking applications requiring autonomous control of multiple ROIs. This methodology additionally provides for reduced ROI readout time and higher frame rates as compared to a predecessor architecture [1] by avoiding external control intervention during the ROI readout process.

I. Introduction

Previously, we reported on a custom camera with real-time software control of single-line readout for acquiring two regions-of-interest (ROIs) [1]. This architecture required intimate coupling between the host controller and the camera logic, with an associated penalty in terms of system operational speed and functional capabilities. In this article, we describe the functionality, architecture, and control methodology of a random access, real-time, event-driven (RARE) camera as part of a real-time target acquisition and tracking platform. The camera implementation uses a Texas Instruments TC237 charge-coupled device (CCD) focal-plane array (FPA) and two TLV987 signal processors [4]. Control of the imager and signal processors is via custom logic in a field programmable gate array (FPGA) that accepts user commands and provides region-of-interest pixel data to a host tracking processor. A message-passing paradigm is used to provide real-time imager control without knowledge of detailed imager operation.

In the remainder of this section, we discuss the system functional goals and RARE camera requirements. Section II describes the RARE camera hardware architecture. Section III discusses the ROI readout mechanics. Section IV presents information regarding an embedded-pixel data preprocessing

[1] In Situ Technology and Experiments Systems Section.

[2] Communications Systems and Research Section.

[3] Mobility Systems and Technology Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

module. Section V details the camera interface (I/F) and software control. Section VI presents timing benchmarks of camera operation for single- and dual-ROI operation. Section VII exhibits camera characteristics for ROI operation. Conclusions are presented in Section VIII.

A. Camera Requirements

The primary motivation for this camera development is to realize an adaptive sensor mechanism as part of a platform for real-time autonomous acquisition and tracking applications [2]. Such a platform requires both a sensor and a control philosophy that provides real-time adaptation of the sensor based on target characteristics and dynamics and environmental conditions. To achieve this goal requires a camera capable of frame rates of several hundred to several thousand frames per second with operating parameters that can be adjusted on a per-frame basis.

High frame rates with adaptive imager control are achieved with a conventional CCD by reading out only the pixel regions of interest and discarding all other pixels. This mode of operation required the development of a customized local controller for the CCD imager to provide a tightly coupled mechanism for imager operation. Configuration of the controller is handled by a host tracking processor, discussed in Section II, that loads initialization and tracking parameters into the controller to define imager operation. The initialization parameters are needed for defining the start-up mode of the controller, and the tracking parameters define detailed operation of the imager during acquisition and tracking operations.

The functional evolution of the RARE camera is presented in Table 1. Version 1.0 used software control of low-level CCD operations as a first implementation but required tight coupling of the camera with the host tracking processor. This release of the camera was discussed previously [1]. Version 2.0 represents the current state of the RARE camera development. Version 3.0 is the desirable goal for the camera development to achieve autonomous operation with little or no intervention from a host tracking processor. In this article, we discuss the Version 2.0 camera development effort as a migration path to implementing Version 3.0.

II. Hardware Architecture

Release 2.0 of the RARE camera is an expanded version of the release 1.0 design [1]. It is used as part of a real-time target-tracking apparatus for free-space optical communications and non-invasive eye tracking [2] and provides simplified high-level control of low-level camera operation on an intra-frame basis. The architecture of the release 2.0 RARE camera is shown in Fig. 1.

This figure illustrates the three-component system of the RARE camera, consisting of a custom imager card with a low-voltage differential signaling (LVDS) cable assembly, a commercial off-the-shelf (COTS) field-programmable gate array (FPGA) card, and a commercial processor (e.g., personal computer). The imager card was designed with a Texas Instruments TC237 CCD imager chip [3] with two TLV987 signal processor chips [4]. Two 987 processors were required to handle the dual-pixel-stream output capability of the TC237 CCD. Each processor accepts an analog pixel stream and provides transistor–transistor logic (TTL)-level output signals to the custom LVDS cable. The cable assembly provides single-ended TTL-level input–output (IO) signals to the CCD card and the FPGA card but runs differential signals through a pair of small computer system interface (SCSI) cables to allow for high-speed strobe operation over several feet of cable. The FPGA card is a TransTech PMCFPGA-01 card with a 300 kilo-gate Xilinx XCV300E FPGA for the low-level controller of the CCD imager. The host tracking processor is a general-purpose computer with a 32-bit peripheral component interconnect (PCI) bus used to provide FPGA control parameters and read camera status and pixel data from the FPGA card.

A. FPGA Registers

The hardware architecture of the RARE camera shown in Fig. 1 utilizes the custom FPGA controller to achieve real-time, event-driven operation. This philosophy is facilitated by defining registers from

Table 1. RARE camera version definitions.

Feature	Version 1.0 camera (TC237 with new FPGA)	Version 2.0 camera (new FPGA)	Version 3.0 camera (new FPGA)
Interfaces	I/O to PCI	I/O to PCI	I/O to PCI
Maximum frame size	256 × 256	Full frame	Full Frame
Exposure control	Via frame rate	User-programmable via hardware	User-programmable via hardware
Frame-rate control	Software time base	User-programmable via hardware	User-programmable via hardware
Pixel processor (987) programmability	Required	Required	Required
Windowing	Software	User-programmable via hardware	User-programmable via hardware
DMA	No	Yes	Yes
OS	Real-time operating system (RTOS)	RTOS/Windows	RTOS/Windows
Auto-gain	No	No	Yes
Time-tagging[a]	No	No	Yes
Pixel outputs	One	Two	Two
Multiple operating modes[b]	No	Yes	Yes
Large image buffer for data storage	No	No	Yes
Pulsed time-based output[c]	No	No	Yes
Sequenced commands	No	No	Yes
Preprocessing of pixel data	No	Limited	Yes

[a] The FPGA maintains time-based information.

[b] For streaming video versus single-image capture. The configuration is defined by setting bits in the control register of the FPGA.

[c] The FPGA provides a single output that pulses at some prescribed rate.

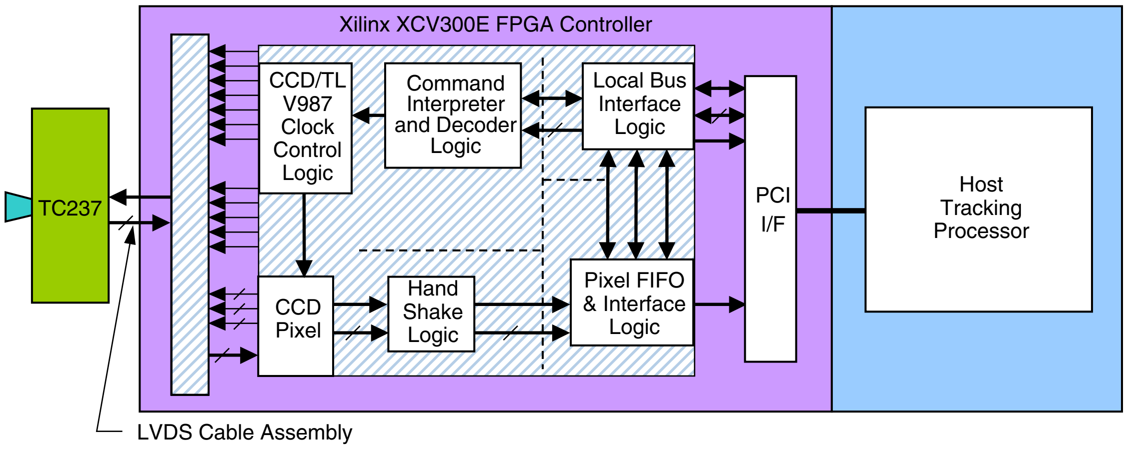
Xilinx XCV300E FPGA Controller
TC237
CCD/TL V987 Clock Control Logic
Command Interpreter and Decoder Logic
Local Bus Interface Logic
CCD Pixel
Hand Shake Logic
Pixel FIFO & Interface Logic
PCI I/F
Host Tracking Processor
LVDS Cable Assembly

Fig. 1. Tracking system.

a user perspective to include such parameters as ROI size and location, integration time, and frame rate. By customizing FPGA controller operation via these registers, the RARE camera provides low-level CCD imager control based on high-level dynamic requirements. Modification of these registers typically is done on a per frame basis, but some registers also can be modified on an intra-frame basis, such as the multiple-ROI readout discussed in Section II.B.

The control scheme is implemented through two types of registers, denoted as direct and buried registers. The former are memory-mapped locations that can be written/read directly via a PCI device. The latter are not directly accessible via the PCI but can be configured using RARE camera commands. These commands consist of two-, four-, or six-word sequences, with a word defined as 32 bits. The number of words used in a command depends on the amount of data and address information needed to load a particular buried register or group of buried registers in the FPGA.

1. FPGA Direct-Register Definitions. The directly accessible registers are listed below. A brief explanation of register functionality is given along with the PCI address offset, the register size in bits, and write/read capability. These registers typically define the FPGA configuration or initiate an action by the camera.

> TEST: 64-bit readable register with PCI offset 0×40 hex (hexadicimal). This register is a dual-function location used to read test points within the FPGA or to read pixel data from the CCD. The multiplexing logic used to define the mode for this register is set up via the CONTROL register.
>
> CONTROL: 64-bit control and status register with PCI offset 0×48 hex. This write/read register is used for FPGA configuration control.
>
> PORT0: 32-bit write-only register with PCI offset 0×50 hex. This register location is used to write two-, four-, or six-word command sequences to configure the buried registers.
>
> CCDREQ: 32-bit write-only register with PCI offset 0×70 hex. A write to this register requests that the CCD controller begin a frame transfer or ROI readout depending on the mode selected.
>
> CCDSTAT: 32-bit read-only register with PCI offset 0×74 hex. The contents of this register provide information pertaining to the state of the controller in either frame-transfer or ROI-readout modes. In frame-transfer mode, this register provides a flag indicating the end of a frame-transfer operation. In ROI readout, this register provides access to the counter that tracks the number of available pixels in the pixel first in, first out (FIFO) and a flag indicating completion of a ROI readout operation.
>
> CCDSTRST: 32-bit write-only register with PCI offset 0×78 hex. This is used to reset individual bits in the CCDSTAT register.

2. FPGA Buried-Register Definitions. The buried registers are used to define the detailed operation of the CCD control logic. These registers are accessed via the PORT0 (i.e., command port) location and are written to using two-, four-, or six-word command sequences, with a word defined as 32 bits. The buried registers are listed below with a brief explanation of register functionality.

> CREG: 32-bit write-only register used to define the FPGA control mode. This register currently defines the frame transfer versus the ROI readout mode.
>
> FRAME_REG: 32-bit write-only register used to define the dwell time between two frame-transfer operations. This register controls the frame rate of the camera.
>
> INT_REG: 32-bit write-only register used to define the integration time. The value in this register can be equal to or less than the value in FRAME_REG.

SREG: 16-bit write-only register used to load the registers (discussed below) of the TLV987 signal processors.

WIDTH_REGA: 10-bit write-only register used to define the width of the ROI. Currently only one set of ROI registers is used. When additional ROIs are folded into the controller, the "A" set of registers will pertain to one of N register sets.

X_ORGA: 10-bit write-only register used to define the horizontal starting position (in pixels) of the "A" ROI.

LN_SKIPA: 10-bit line-skip register used to define how many lines to scroll between line readout operations within the ROI.

HEIGHT_REGA: 10-bit write-only register used to define the height of the "A" ROI.

Y_ORGA: 10-bit write-only register used to define the vertical starting position (in pixels) of the "A" ROI.

3. TLV987 Video Signal Processor Registers. The TLV987 is an analog signal processor with an onboard analog-to-digital converter (ADC) used to process one pixel stream of the CCD imager. The signal processor contains a correlated double sampler, 10-bit 27-Megasample/second analog-to-digital converter, programmable gain and dark offset control registers, and a serial interface for register configuration [4]. Software access to the control registers is accomplished via a three-wire interface consisting of the serial data, clock, and a chip select line per the TLV987 processor [4].

Release 2.0 allows for setting the TLV987 internal registers to operate in a user-defined mode as dictated by operational circumstances. These registers are loaded as buried registers using 6-word commands. The serial register formats and default modes are given in Table 2.

B. FPGA Implementation Details

Camera operation consists of four fundamental operations. They include the dwell time, the frame-transfer operation, line scrolling, and ROI readout. The minimum time required to read out a ROI is defined as the sum of the times for the frame-transfer, the line scrolling, and the ROI readout operations.

Figure 2 shows the sequence of these operations and their associated timing signals. The mode flag is a bit in the control register of the FPGA, and its state defines the operation of the camera in response to a request from the host tracking processor. A request to the camera is issued by writing the CCDREQ register and results in pulsing the flag, CCDReq, shown in Fig. 2. The response of the camera will be a frame transfer or line request, depending on the state of the mode flag. If the mode flag is set, a frame transfer will commence, as indicated by the frame-transfer flag. If the mode flag is low, a combination of pixel line requests and line readout operations will be performed to commence a ROI transfer, as denoted by the line valid flag in Fig. 2.

1. User-Programmable Frame-Rate and Integration-Time Control. The frame-rate control function uses the FRAME_REG register to store a count value that corresponds to the dwell time and is defined as the number of 20-MHz clock cycles between frame-transfer operations. This register is used to initialize the dwell-time down counter at the completion of the current frame transfer. The next frame transfer is automatically requested by the FPGA controller when the down counter goes to zero. During a frame-transfer operation, this counter is held at zero until the frame transfer has completed. This functionality provides the host tracking processor with a mechanism to set the integration time and the frame rate of the camera.

A provision is included to allow for independent control of the integration time with a given dwell time. The INT_REG stores the integration-time count. This count is compared to the dwell-time count

Table 2. TLV987 signal processor registers.

Register	Address	Default value	Description
Control	0	000 hex	Defines TLV987 configuration
PGA	1	0 dB	Programmable gain; least-significant bit (LSB) $\cong$ 0.1-dB gain
User digital-to-analog converter (DAC)1, DAC2	2, 3	00 hex	Programmable analog outputs
Course offset DAC	4	±00 hex	Pre-PGA black and offset correction
Fine 0ffset DAC	5	±00 hex	Post-PGA black and offset correction
Optical black level	6	040 hex	Black-level set point
Optical black calibration	7	003 hex	Auto black-level definition
Test	8	003 hex	Test point definition

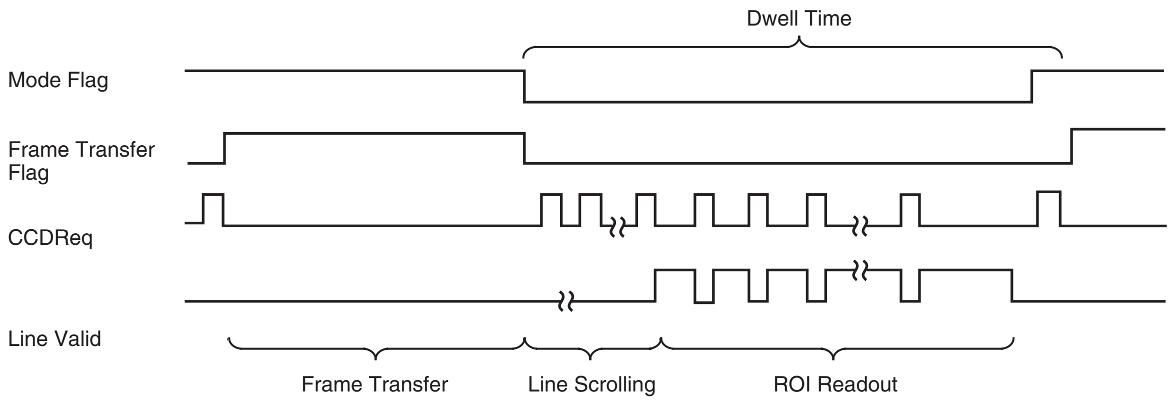

Fig. 2. Camera operations.

and generates a reset pulse to the CCD imaging area when they are equal. This value must be equal to or less than the dwell-time count to be meaningful. Values larger than the dwell count will have no effect on imager operation.

If these registers are set to zero, a frame transfer will occur only when requested by the host tracking processor. This case will result in variable integration time and essentially operates the camera in a single-shot mode. If these registers are set to the same non-zero value, then the camera will request frame transfers autonomously, and the dwell time will equal the integration time. In this mode, the integration time can be shortened with respect to the dwell time by setting the integration-time register to a smaller value than the dwell-time register. In either case, this mode is referred to as the autonomous frame-transfer mode.

III. ROI Readout Mechanics

The Version 1.0 implementation required intervention by the host tracking processor to read out only the pixels of interest. Version 2.0 moved this functionality into the FPGA. The ROI function was implemented in the FPGA controller to automate the process of selective pixel readout from the TC237 CCD. Pixel data are read out from the TC237 CCD based on ROI size and location information stored in the FPGA and are deposited into the pixel FIFO of the FPGA.

A. ROI Definition

A ROI within the field of view (FOV) of the CCD is defined by four ROI configuration registers consisting of the (x, y) origin, X_ORGA, Y_ORGA and the width and height information $(\Delta x, \Delta y)$, WIDTH_REGA and HEIGHT_REGA. Each of these registers is 10-bits wide to allow for ROIs positioned anywhere within the 684 columns and 500 rows of the TC237 FOV and is sized from 1 pixel to the maximum number of pixels in a dimension. These registers are accessed with 8 bits of address information. The format for writing into these registers consists of a 6-word command. The first word is null, the second word defines the command, the third word is null, the fourth word contains the address byte for the ROI register set, and the fifth and six words contain the origin and size information of the ROI as 10-bit registers. These registers are loaded prior to requesting a ROI readout operation and are used to load counters within the FPGA controller for ROI readout. They can be updated when ROI readout is not in progress and a new ROI size or location is required.

In addition to the ROI size and position registers, an additional 10-bit register per window is defined for line skipping within a ROI. Writing to this register requires a 6-word command. The register defines the number of lines to skip between lines read out in the ROI. This register can be nominally set to zero after a reset to allow for readout of the window without line skipping. A second companion register also is planned to define the number of columns to skip between pixels on a line. This column-skip register would be accessed with the same 6-word command used to initialize the line-skip register.

The ROI registers are used in two ways. The first defines which lines and pixels per line are read out from the imager into the pixel FIFO of the FPGA. The second provides a mechanism for tagging pixels in terms of their associated ROI(s). The reason for this distinction is to provide a mechanism to handle pixels in overlapping ROIs. Readout of such pixels from the CCD is done only once, but pixels in overlapping regions need to be associated with multiple ROIs. By using flags from the registers for a ROI, a flag can be defined to map a pixel into the associated ROI. This capability is not required with a host tracking processor but is useful for an autonomous system that does not have the tracking processor.

B. Single-ROI Readout

A request for a ROI readout is initiated by the host tracking processor (after defining the ROI registers) and proceeds autonomously. The FPGA controller provides pixel FIFO status information from an up/down counter that tracks write/read operations to the pixel FIFO. The count value is accessed by the host tracking processor via the CCD status register, CCDSTAT, and is used to determine the timing and quantity of pixel data available for readout. For large-format ROIs, the FPGA controller additionally controls the rate at which pixel data are loaded into the onboard pixel FIFO. This mechanism is used to avoid losing pixel data if the host tracking processor cannot retrieve the pixel data in a timely fashion.

In the autonomous frame-transfer mode, the ROI readout time may exceed the programmed dwell time. In this instance, the FPGA controller periodically resets the CCD imaging area, with the time between reset pulses defined by the dwell time. This mechanism allows for variable ROI readout timing due to indeterminate timing on the part of the host tracking processor but maintains a fixed dwell/integration time. When the ROI readout operation is complete and the FPGA controller is switched back to the frame-transfer mode, the camera will begin the next frame transfer after the current dwell period has expired. So, ROIs can be read out at whatever rate is required (but not faster than the frame rate) without affecting the integration time.

C. Multi-ROI Operation in Software

The host tracking processor loads ROI parameters into the FPGA controller to define the ROI size and location. These parameters are used to scroll through unwanted lines and unwanted pixels per line until the ROI is reached. They additionally define the number of pixels per line and the number of lines to read out for the ROI.

Dynamic adjustment of these parameters can be done on a per-frame or intra-frame basis to allow for enhanced system adaptation. They can be used to define a ROI for one or more frames, or they can be adjusted within one frame to allow for multiple ROIs within a single frame. This case is depicted in Fig. 3 for two ROIs.

When readout of a ROI(s) is required, the tracking processor sets the mode of the FPGA controller to ROI readout and requests pixel data as defined by the ROI parameters previously loaded. The tracking processor then polls the FPGA controller for available pixel data to initiate the ROI readout from the pixel FIFO in the FPGA. If more than one ROI per frame is required, the tracking processor can load new size and location parameters for the next ROI *without requesting a new frame transfer.* The vertical location parameter of the next ROI is defined relative to the last line of the current ROI and denoted by the inter-window scroll area shown in Fig. 3.

This methodology also allows for ROIs that overlap or share common rows of pixels with or without vertical separation. The primary difference as compared to the case above is in the definition of the ROI. For the case of common lines of pixels between ROIs, the tracking software must read out three different regions corresponding to the two areas where the ROIs do not share common rows of pixels and an additional third region containing pixels from both ROIs. This additional region must read out pixels for both ROIs and will have a width parameter defined by the left edge of the left-most ROI and the right edge of the right-most ROI. This methodology is illustrated in Fig. 4 for two ROIs and is applicable to $N(> 2)$ ROIs.

D. Multi-ROI Readout Implementation

The only input required from the host tracking processor for ROI readout is the origin and size information. A ROI is ignored if Δy is set to zero. Single-ROI pixel readout uses line and column request logic to determine the lines and pixels per line to read out. The origin and size information loaded into the ROI registers is used to initialize counters that track progress of pixel data read out from the CCD. The line request logic is used to perform line scrolling and line readout when prompted by a ROI request from the host tracking processor. This logic uses the y and Δy information to determine the active lines of the ROI within the CCD FOV to generate a ROI active flag. The active state of this flag coupled with subsequent line requests results in generation of the line valid flag to denote the readout of pixel data

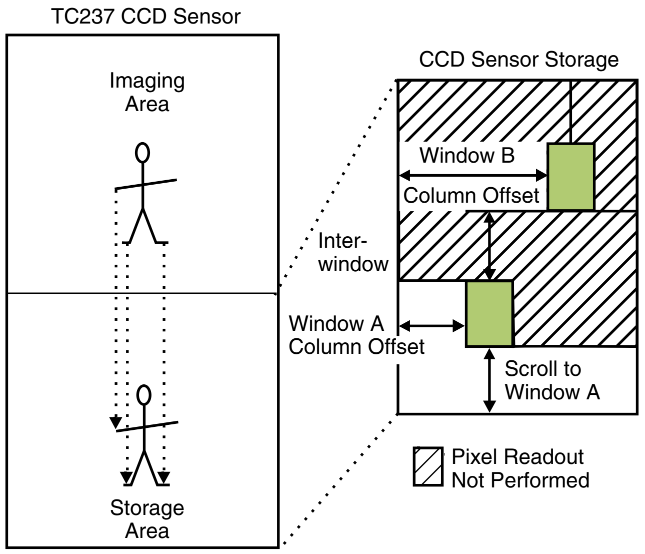

Fig. 3. Frame transfer and ROI readout.

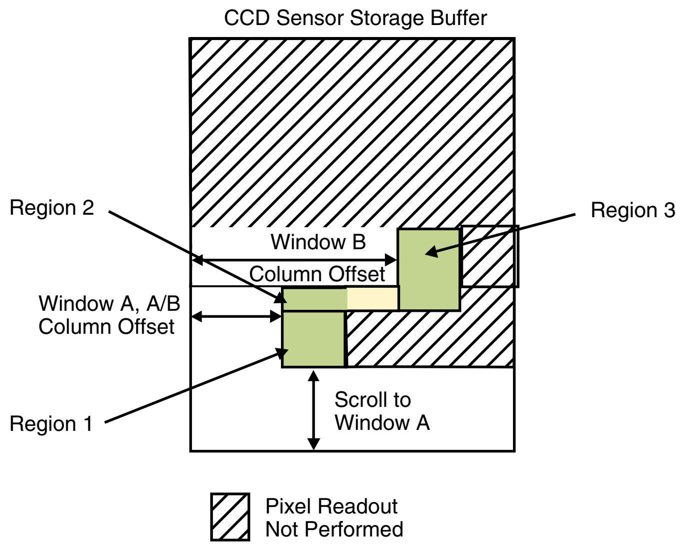

Fig. 4. ROI readout with common rows of pixels.

from the CCD. The column request logic uses the x and Δx information to modulate the line valid flag to define which pixels on a line are to be loaded into the pixel FIFO of the FPGA.

The above methodology is extensible to multiple ROIs within the CCD FOV. Two new modules were developed that incorporate the line and column request logic, respectively. Moving the appropriate code into lower-level modules allows for a more modular design amenable to installing multiple ROIs. The current logic allows for two ROIs that can be located anywhere within the CCD FOV and requires two copies of the line and column request modules, as shown in Fig. 5 for $M = 2$.

1. Multiple-ROI Pixel FIFO Management. The FPGA design provides pixel FIFO status information using an up/down counter that tracks the FIFO write/read operations, respectively. This status information provides a real-time readout mechanism of the pixel FIFO while a line of pixels is being written into the FIFO from the CCD. The count value is accessed via the CCD status register by the host tracking processor and is used to determine when data are available for readout. This mechanism is required to avoid the latency needed to fill the pixel FIFO before beginning readout.

The up/down counter status is used as the pixel FIFO flow control mechanism for readout ROI data. The FPGA controller uses this status to determine if the pixel FIFO has enough unused storage space to load a new line of pixel data for the ROI. The host tracking processor uses this status information to determine if data are available in the pixel FIFO. This information is additionally used when reading pixel data via direct memory access (DMA), to determine if a block of pixel data is available in the FIFO. A block transfer is the indivisible DMA operation and requires that at least one block of data be available before allowing the DMA process to initiate or continue.

For a single ROI, this flow control mechanism results in enabling the writing and reading of pixel data based on the up/down counter status. This process is essentially the same for the dual ROIs but with the added condition of maintaining synchronization between the DMA and the pixel FIFO up/down counter regardless of the locations of the ROIs. This additional constraint translates into avoiding emptying the pixel FIFO until all pixel data for the ROIs are read out from the CCD into the pixel FIFO.

In implementing the dual-ROI logic, a loss of synchronization between the FIFO up/down counter and the DMA was encountered for the case of one or more rows of separation between the two ROIs. This effect is due to emptying of the FIFO and reading of additional data from the FIFO that were not real

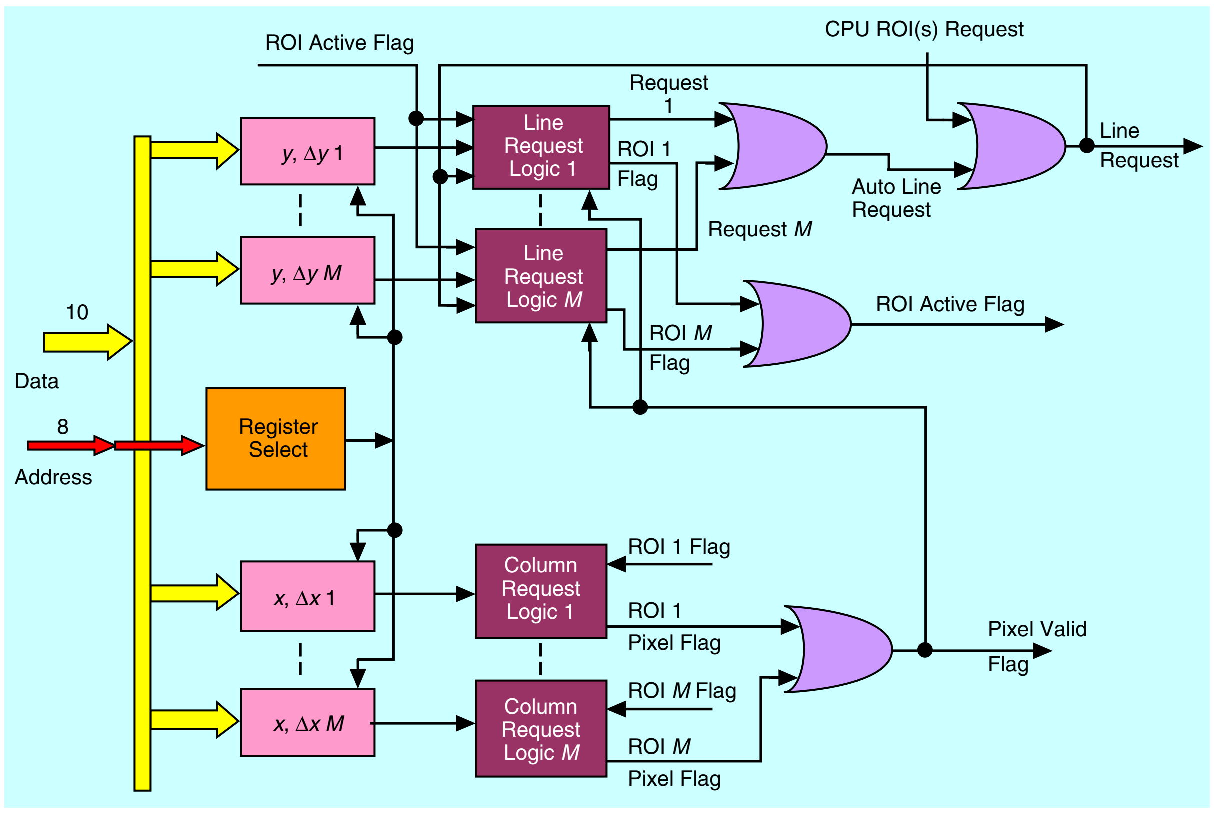

Fig. 5. Generalized ROI readout logic.

pixel data because the "almost-empty flag" for the pixel FIFO is dropped to zero to allow completion of the read operation after the first ROI readout from the CCD has completed. For dual-ROI operation, the DMA must be stalled after completion of the first ROI readout but before the second ROI readout commences if the value of the FIFO up/down counter is less than the DMA block size.

This problem was rectified by defining a new flag that is asserted from the time the ROI readout request is issued by the host tracking processor and is held active until the last line of the last window is read out. This new flag is denoted the "ROI Active Flag" in Fig. 5 and is used to inhibit initiation of a DMA block transfer until at least a block or more of data is available in the FIFO. Draining of the FIFO is allowed only when the last line of the second ROI is read out. This methodology is extensible to more than two ROIs.

IV. Embedded-Pixel Data Preprocessing

A new processing module was added to the FPGA logic for finding features based on programmable intensity thresholds and contrast ratios. This block allows for generating an edge-enhanced image in real time and selecting two features that fall within the programmed thresholds. Two readout modes are defined that allow for recovery of the raw image data and edge-enhanced data in parallel or for only the processed results. The processed results provide feature information within the ROI(s). These results provide a significantly reduced data set while still extracting desired feature information with no time penalty for the results.

This pixel data preprocessing logic provides feature information within the ROI as a means of reducing the computational load on the host tracking processor. The data processing function of intensity-threshold checking entails comparing pixel data to a predefined threshold range and tagging pixels that are within the range. All other pixels are ignored. This function allows for identifying pixels within an intensity range and can be used to quickly tag features in the ROI. The tagging function is accomplished by setting a flag indicating that a pixel(s) falls within the predefined intensity range. On a per row basis, the intensity and location of the pixel with the highest intensity in the feature is stored in the pixel FIFO for read out as part of the processed results. By defining one such pixel per row (if found), a vertical feature within the ROI can be identified.

The function of contrast enhancement is implemented through use of a finite impulse response (FIR) filter within the pixel data path of the FPGA. This logic is used to sense contrast changes that are equal to or greater than a programmable threshold and that are within the intensity range specified by the threshold-checking logic discussed above. On a per row basis, the largest contrast ratio pixel found within the defined parameters is tagged with an additional flag. The position of this pixel within the row of the ROI and its contrast level are stored in the pixel FIFO for readout as part of the processed results. By defining one such pixel per row (if found), a vertical contrast feature within the ROI can be identified.

This processing logic can be used to sense for a bright object, such as a beacon in acquisition mode, to alleviate processing on the host tracking processor during this phase of operation. In this mode, only the processed results would be read back to indicate the position of a beacon and define the initial set of ROI coordinates. Upon defining the ROI position, the intensity flag information can be used to determine the extent of the beacon in order to appropriately size the ROI. If the target is more elaborate than a simple beacon, then the edge-enhanced image could be used to identify the target and provide positional information for ROI manipulation.

V. Camera Interface and Software Control

The RARE camera is accessed as a memory-mapped device via the PCI bus of the host tracking processor, as shown in Fig. 1. This architecture allows for both programming the Xilinx FPGA and communicating with the FPGA as a CCD imager controller via the PCI bus. The control software running on the host tracking processor communicates to the camera by writing sequences of 32-bit words to the FPGA controller for register initialization and performance of various operations. A response from the camera is generated by flags denoting the availability of a new frame within the CCD storage area or data in the pixel FIFO of the FPGA. This response mechanism provides the synchronization of the host tracking processor software to camera operation without knowledge of detailed camera operation.

A. Host Tracking Processor Interface

The COTS FPGA card used for the RARE camera contains a master–slave PCI interface for communication between the host processor and the memory and FPGA on the card. This functionality is achieved with a dedicated PCI interface chip that maps the PCI signals to a local data/address bus structure. The low-level details of the PCI interface are described in [5].

The PCI interface chip can be programmed for a variety of interface options. As part of the RARE camera, the card is used only as a PCI slave device capable of 32- or 64-bit transfers. Data transfers can be single-memory read or write operations, or they can be defined as direct memory access (DMA) transfers for high-speed throughput. The DMA operation can be further defined according to the size of an indivisible block transfer, where the complete DMA transfer consists of one or more block transfers. DMA operation is also specified in terms of running in a simplex or half-duplex mode [6]. These options allow for optimizing DMA operation for a given application.

Within the context of the RARE camera, single accesses are used for loading parameters and reading camera status. ROI readout operations use the simplex DMA mode with a block size of sixty-four 64-bit words to reduce pixel readout time. The block size comes into play by allowing for stalling of the DMA operation on block boundaries if not enough pixel data are available for the next block transfer. This mechanism is used as the rate-limiting mechanism by the tracking processor to read out data as fast as the CCD imager can supply data without under-flowing the pixel FIFO in the FPGA. This constraint comes from the fact that the CCD card currently has a maximum data rate of 20 megapixels per second for dual-pixel-stream output. A pixel is 10 bits and is placed in the 2 low-order bytes of a 32-bit word. The PCI DMA operation can sustain 33 megawords per second, or correspondingly, 33 megapixels per second.

The use of DMA transfers provides for higher throughput over single-location accesses by providing dedicated PCI bus operation during a block transfer. The use of the stall mechanism additionally allows for simultaneous reading of pixels from the pixel FIFO in the FPGA while pixels are being written into the FIFO from the CCD storage area. This scheme effectively limits the camera throughput to the CCD imager readout rate with a small latency to load the first block of pixel data into the FIFO.

B. Software Flow

A flow chart of the host tracking processor software is shown in Fig. 6. The basic functions of the tracking processor are to provide control information to the camera and to read and process pixel data from the camera. The control is in the form of initialization and command sequences used to operate the camera. The results from the processed data are for target recognition and tracking via adjustment of the window size and location parameters.

The flow begins by initializing the TLV987 signal processor chips and setting the FPGA controller for single- or dual-pixel-stream operation with the mode flag set for frame transfers. A request for a frame transfer is issued, followed by initialization of the DMA process used to read out the first ROI. After completion of DMA initialization, the CCDSTAT register is polled to determine when the frame transfer is complete. Upon sensing this event, the mode is switched to ROI readout, and the DMA

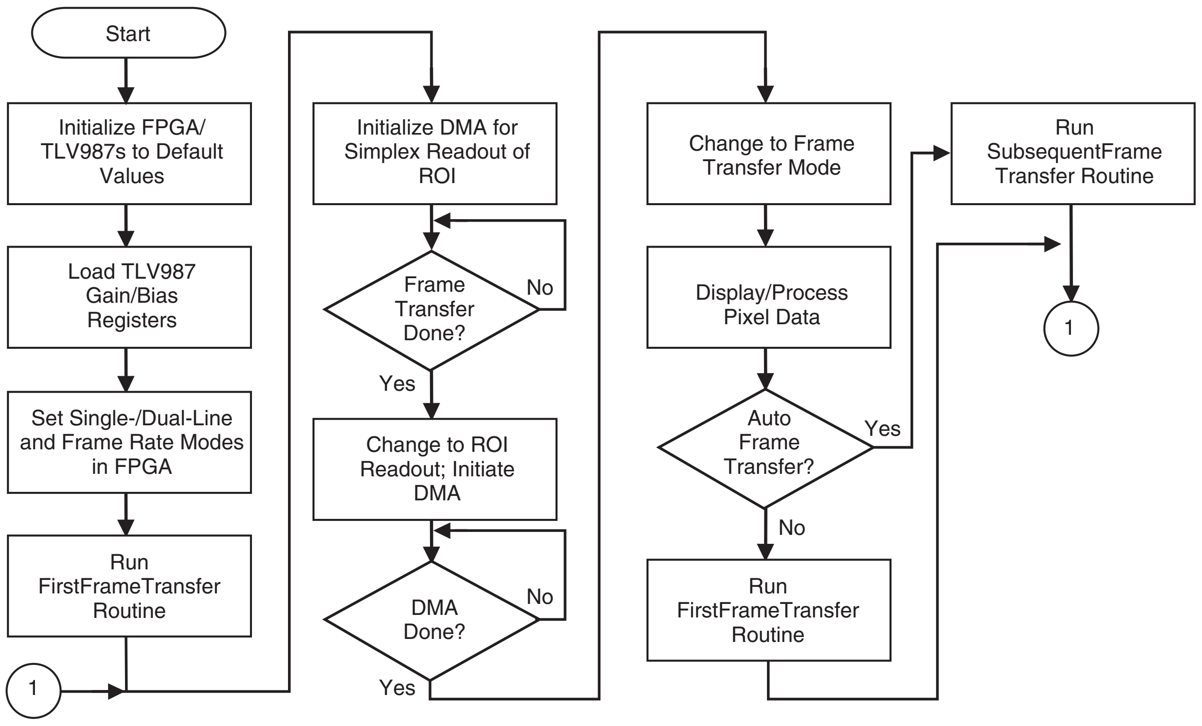
Start
Initialize FPGA/ TLV987s to Default Values
Load TLV987 Gain/Bias Registers
Set Single-/Dual-Line and Frame Rate Modes in FPGA
Run FirstFrameTransfer Routine
1
Initialize DMA for Simplex Readout of ROI
Frame Transfer Done?
No
Yes
Change to ROI Readout; Initiate DMA
DMA Done?
No
Yes
Change to Frame Transfer Mode
Display/Process Pixel Data
Auto Frame Transfer?
Yes
No
Run FirstFrameTransfer Routine
Run SubsequentFrame Transfer Routine
1

Fig. 6. Tracking processor control software flow chart.

operation commences. After completion of the DMA operation, the controller mode is switched back to frame-transfer operation, and the retrieved pixel data are displayed and/or processed. If the autonomous frame-transfer mode is being used, the DMA process is immediately initialized in preparation for the next ROI readout. If the manual frame-transfer mode is being used, a new frame-transfer request is issued before initializing the next DMA operation. This sequence of frame transfer followed by ROI readout is maintained for the duration of a tracking experiment.

C. Software Control

The tracking processor software controls the RARE camera by sending commands to the FPGA card. They are a combination of writing to the direct or buried registers in the FPGA, followed by accessing of the CCDREQ register. The format for writing into the buried registers consists of six 32-bit words: the first word is null; the second word defines the command; the third word is null; the fourth word contains the address of the register or set of registers; and the fifth and six words contain the data for the register(s).

The host tracking processor uses a variety of register write operations to define higher-level commands. They provide the primary means of camera control and synchronization. The various commands are listed and defined below.

InitializeFPGA: loads the registers in the FPGA controller and TLV987 signal processors with default values.

InitializeGainControl: sets the gain registers in the 987 signal processors.

DeletePF: Releases the FPGA device.

FirstFrameTransfer: initiates the frame-transfer operation by setting the FPGA controller in the frame-transfer mode and requesting the first frame-transfer operation.

SubsequentFrameTransfer: sets the FPGA controller to the frame-transfer mode to resume frame-transfer operations.

InitializeSingleLineRead: sets the line scroll register to load one line of pixels into a shift register for single-line readout.

InitializeDualLineRead: sets the line scroll register to load two lines of pixels into the shift registers for dual-line readout.

FetchImageData: gets an image from the CCD.

GetFIFOPixel: gets a single pixel from the pixel FIFO.

GetDualPixels: gets pixels from the FIFO using DMA readout with two-pixel intensity values stored in a 64-bit FIFO location.

GetFirstLinePixel: reads out the odd field of pixel data.

GetSecondLinePixel: reads out the even field of pixel data.

SplitValue: divides a 10-bit pixel value into two parts: the upper part contains the 8th and 9th bits; the lower part contains bit 0 to bit 7.

SetCCDControl: sets a TLV987 register to a specific value. The TLV987 register can be the control register, the CoarseDAC register, or the CCDGain register.

Frame Rate/Exposure Control: defines the amount of time the CCD spends collecting charge for a particular image and the rate at which frame-transfer operations occur. Exposure control allows for clearing of the imaging area without performing a frame transfer.

VI. ROI Timing Benchmarks

A series of benchmark experiments was performed on the RARE camera to characterize the actual frame rate for a single ROI per frame. The results show the achievable frame rates for different sizes and locations of ROIs within a frame. These results are compared with the maximum theoretical frame rates, which include the CCD frame-transfer operation, ROI size and location definition, data transfer from the CCD to the pixel FIFO in the FPGA, and transfer of pixel data to DMA memory in the host tracking processor.

A. Experimental Setup

The benchmark experiments were run on two different platforms. One platform consisted of an IBM-compatible PC with an AMD 2000+ central processing unit (CPU) and 512-MB dynamic random access memory (DRAM) for the host tracking processor. The operating system (OS) is Windows 2000, SP2. The FPGA card driver is Version pf_1.3r0 from Transtech and is commercially available. The other platform used a Synergy VGM5 power PC processor running VxWorks, Version 5.4.2.

The customized firmware in the FPGA for local camera control is Version pf1t_81v1b and supports the functionality given in Table 1. The TC237 CCD is operated with dual-pixel-stream outputs with each stream running at 10 megapixels per second. Dual-pixel-stream readout refers to transfer of pixel data from the CCD storage area to the FPGA pixel FIFO. Readout of pixel data from the pixel FIFO to the host tracking processor memory uses DMA accesses. The experimental results do not account for time needed to process or display the pixel data.

The autonomous frame-transfer mode is disabled in the experimental trials so that frame0transfer operations occur only when requested by the host tracking processor. Thus, the dwell and integration times are governed by the ROI readout time. The frame-transfer time for firmware Version pf1t_81v1b is 100 microseconds.

B. Single-ROI Experimental Results

The experimental and theoretical results are presented in Figs. 7 and 8. The measured results are presented by the solid curves and the theoretical predictions by the dashed curves. The horizontal axis represents the ROI size or location in pixels, and the vertical axis represents the rate at which images are received from the CCD. For both experiments, the ROI has an $N \times N$ aspect ratio.

Figure 7 shows a family of curves, where each curve represents a particular origin position for the ROI. The origin corresponds to the upper-left corner of the ROI and is the location of the first pixel read from the CCD. The curves in this figure correspond to ROI origins located at pixel positions (0,0), (100,100), (200,200), (300,300), and (400,400). The shape of a curve shows the change in frame rate as a function of the ROI size for a given origin position.

The uppermost curve in Fig. 7 represents the theoretical frame rates for ROIs with origins located at pixel (0,0). The theoretical rate is calculated by adding together the time for all events needed to perform frame transfer and ROI readout. For a 10×10 ROI, the theoretical frame rate is approximately 8.5 kHz. For a 100×100 ROI with an origin at (0,0), the theoretical rate is 1.4 kHz. The corresponding experimental numbers are 6.4 kHz and 1.1 kHz.

Figure 8 shows a family of curves, where each curve represents a particular $N \times N$ ROI size. The shape of each curve shows the change in the frame rate as a function of the ROI origin position. The ROI origin is allowed to move along a diagonal line such that the horizontal and vertical displacements of the origin are equal. The curves shown in this figure are for ROIs of size 10×10, 50×50, and 100×100.

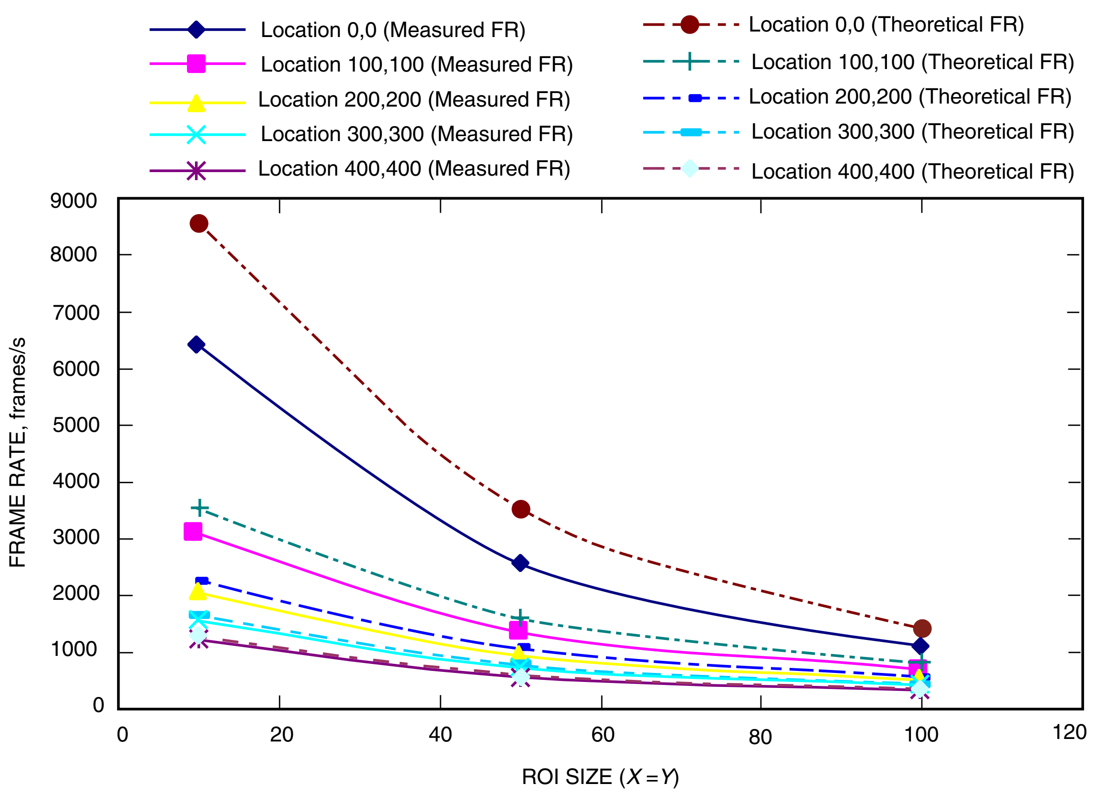

Fig. 7. CCD (DMA) frame rate (FR) versus ROI size summary.

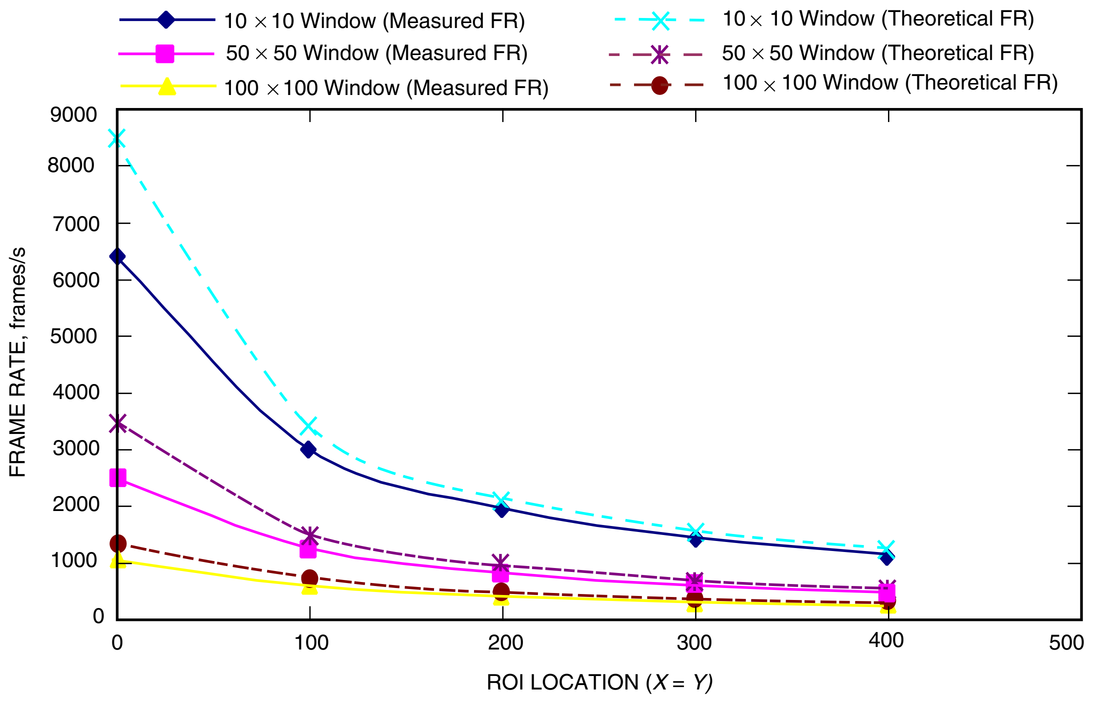

Fig. 8. CCD (DMA) frame rate versus window location summary.

The uppermost curve in Fig. 8 shows the theoretical frame rates for 10×10 ROIs. The corresponding experimental curve is the uppermost solid line. For the origin position of (0,0) and (400,400), the theoretical rates are 8.5 kHz and 1.2 kHz, respectively. The corresponding experimental numbers are 6.4 kHz and 1.1 kHz.

The experimental frame rates are approximately 75 percent of the theoretical rate limits. The difference between the theoretical and measured frame rates is due to several factors. The first is a 20-microsecond delay that was measured between the end of writing the ROI data into the FPGA pixel FIFO and the next frame-transfer request. This delay is due to the time needed to complete the DMA readout from the pixel FIFO and to switch to the frame-transfer mode and request the next frame-transfer operation.

The predominant portion of this 20-microsecond delay is due to the DMA completion time. This time always will be present, due to the need to rate limit the DMA readout as a means of preventing FIFO underflow. The FPGA control logic stalls the DMA operation when the number of pixels in the FIFO drops below the DMA block size. This condition persists until the ROI data have finished loading into the pixel FIFO. Once completed, the DMA is allowed to read all data from the FIFO. The 20-microsecond delay can be reduced by dropping the threshold, but this amount of time is small compared with the 100-microsecond frame-transfer time. Its effect is most notable when the ROI origin is located at (0,0).

The second factor is variable time delay due to a discrepancy in the theoretical line-request timing as compared with actual hardware operation. This delay is due to the use of a 1-microsecond time delay per line request in the theoretical predictions. The experimental results reflect the use of a longer line-request period of 1.25 microseconds due to adjusting of the storage-area and shift-register gate strobes. The use of longer strobe times was investigated to assess effects on a bright-band phenomenon (discussed in Section VII) and results in an additional delay proportional to the number of lines that need to be scrolled before reaching the first line of the ROI and the number of lines requested during ROI readout. The effect of this delay will be most notable for ROIs located in the lower-right quadrant of the CCD FOV.

The third factor is a variable time delay due to the operating system and application software running on the host tracking processor. This delay is not a major factor in the experimental results because images were not processed for this case. This delay will become more noticeable when images are to be processed and the application software requires more than one frame time to process the data. Additional overhead may be incurred due to other processes running on the tracking processor that may result in skipping a frame. The following results do not attempt to quantify these effects.

C. Dual-ROI Benchmark Results

The dual-ROI methodology was benchmarked to determine the achievable frame rates for this scheme. The "home" position of two 11×12 ROIs is defined by origins (320,236) and (320,248). These locations place the ROIs in the same columns with a one-row separation and at the center of the CCD FOV. The top ROI is allowed to move throughout any portion of the upper half of the CCD FOV, and the bottom ROI moves in an opposite sense throughout the bottom half of the FOV. The ROIs are moved in opposite directions to emulate the operational mode required in the acquisition and tracking platform. The reported frame rates include the time needed to perform frame transfers and read out the ROI data only.

Tables 3 through 6 (and the corresponding Figs. 9 through 11) show that frame rates of 900 to 1100 Hz are achievable for dual-pixel-stream operation. As can be seen from Table 4, there is no effect when shifting the ROIs horizontally. The reason for this result is that the total number of pixels read out from the CCD is constant. Consequently, the only parameter that affects the frame rate is the position of the last row of the bottom ROI. The reason for this result is that the only variable for this operational mode is the number of line-scroll operations performed, and the aggregate number of line scrolls increases as the

Table 3. Dual-ROI timing with vertical shift.

Dual-window 11 × 12			Single line		Dual line	
Window origin x	Window 1 origin y	Window 2 origin y	Frame time, ms	Frame rate, Hz	Frame time, ms	Frame rate, Hz
320	236	248	1.30	769	0.87	1149
320	196	288	1.35	740	0.91	1099
320	156	328	1.40	714	0.96	1041
320	116	368	1.45	689	1.01	990
320	76	408	1.49	671	1.06	943
320	36	448	1.54	649	1.11	900

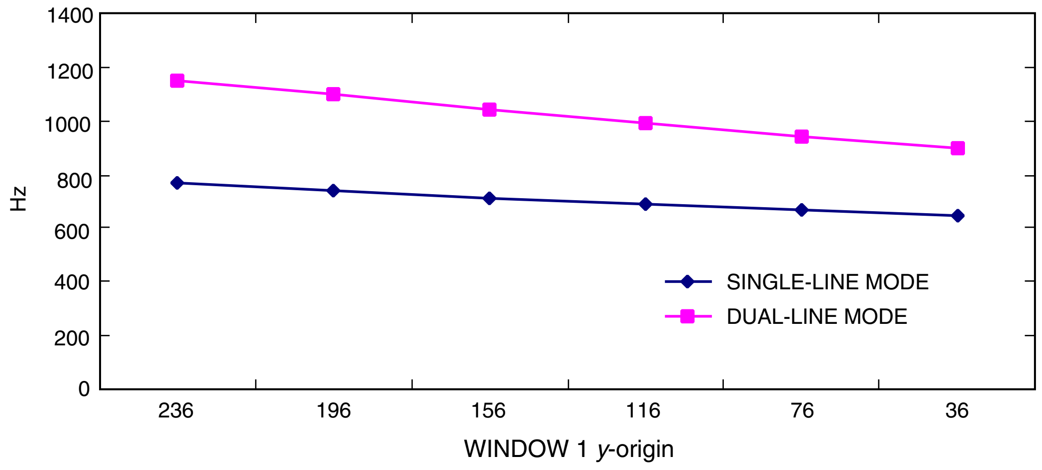

Fig. 9. Dual-ROI timing with vertical shift.

Table 4. Dual-ROI timing with horizontal shift.

Dual-window 11 × 12				Single line		Dual line	
Window 1 origin x	Window 1 origin y	Window 2 origin x	Window 2 origin y	Frame time, ms	Frame rate, Hz	Frame time, ms	Frame rate, Hz
320	236	320	248	1.30	769	0.87	1149
240	236	400	248	1.30	769	0.87	1149
160	236	480	248	1.30	769	0.87	1149
80	236	560	248	1.30	769	0.87	1149
0	236	640	248	1.30	769	0.87	1149

Table 5. Dual-ROI timing with diagonal shift; window 1 in upper-left quadrant.

Dual-window 11 × 12				Single line		Dual line	
Window 1 origin x	Window 1 origin y	Window 2 origin x	Window 2 origin y	Frame time, ms	Frame rate, Hz	Frame time, ms	Frame rate, Hz
320	236	320	248	1.30	769	0.87	1149
260	196	380	288	1.35	740	0.91	1099
200	156	440	328	1.40	714	0.96	1041
140	116	500	368	1.45	689	1.01	990
80	76	560	408	1.49	671	1.06	943
20	36	620	448	1.54	649	1.11	900

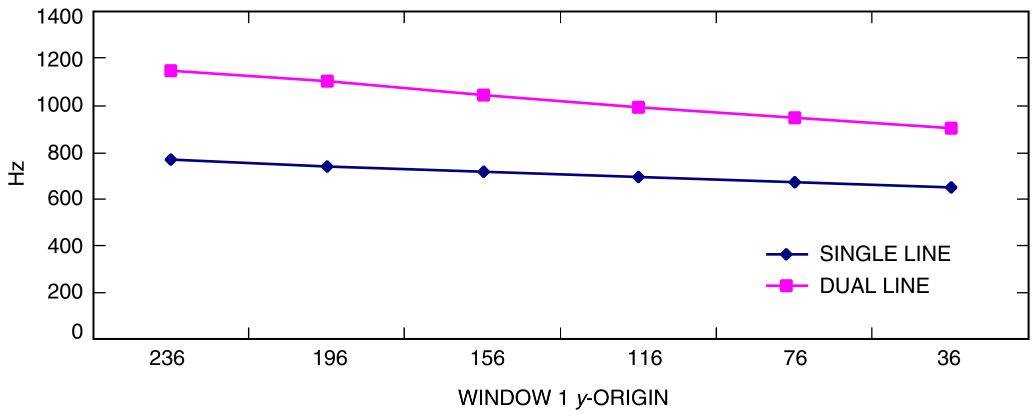

Fig. 10. Dual-ROI timing with diagonal shift; window 1 in upper-left quadrant.

bottom window is moved toward the bottom of the CCD FOV. This effect can be seen when comparing the frame rates in Tables 3, 5, and 6 (and the corresponding Figs. 9 through 11).

An increase in the frame rates exhibited in these tables and figures is possible by using a portion of the CCD FOV. This approach requires the defining of a sub-region of the CCD FOV and the centering of the ROIs in this region. The number of line scrolls and pixel reads is reduced by moving the "home" position of the two ROIs closer to row one, column one of the CCD FOV.

VII. Camera Dual-Line and ROI Characteristics

RARE camera operation as part of a tracking platform requires dual-line ROI operation for high-frame-rate performance. To assess the camera imaging performance in this mode required independent adjustment of the TLV987 pixel strobe timing to minimize the difference in the gain and offset settings of the TLV987 processors for equalized pixel-stream intensities. Following are the discussion of how the TLV987 gain and offset parameters are adjusted for equalization of both flat-field and real scenes and some results for dual-line operation.

A. Flat-Field Image Equalization Results

Here we address the imaging characteristic of the camera with a uniformly lit CCD FOV. For this case, the camera without a lens is illuminated with an integrating sphere. The light level and integration time are selected such that the TLV987 gain and offset parameters fall well within the dynamic range of

Table 6. Dual-ROI timing with diagonal shift; window 1 in upper-right quadrant.

Dual-window 11 × 12				Single line		Dual line	
Window 1 origin x	Window 1 origin y	Window 2 origin x	Window 2 origin y	Frame time, ms	Frame rate, Hz	Frame time, ms	Frame rate, Hz
320	236	320	248	1.30	769	0.87	1149
380	196	260	288	1.35	740	0.91	1099
440	156	200	328	1.40	714	0.96	1041
500	116	140	368	1.45	689	1.01	990
560	76	80	408	1.49	671	1.06	943
620	36	20	448	1.54	649	1.11	900

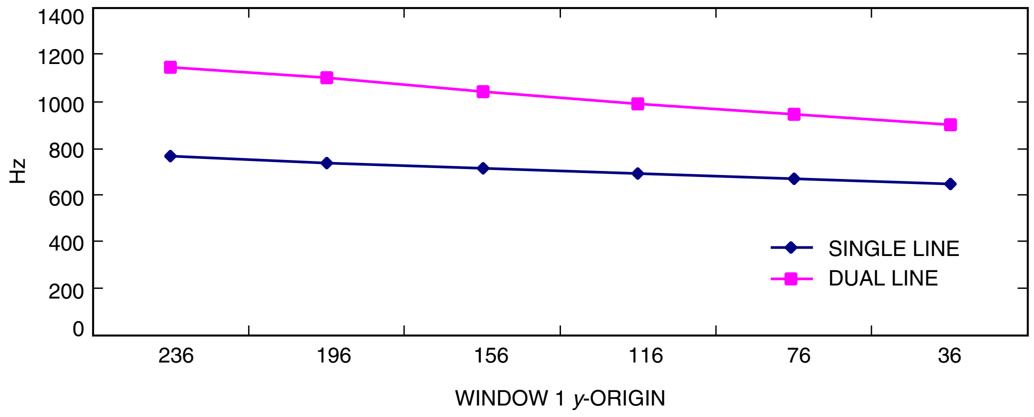

Fig. 11. Dual-ROI timing with diagonal shift; window 1 in upper-right quadrant.

the devices. For the selected TLV987 pixel strobe timings, the mapping of the gains and offsets of the two CCD outputs is given in Table 7; it yields less than 10 percent variation between adjacent rows of pixels in a 256 × 256 area located in the upper-left quadrant of the CCD FOV. The gain is in units of decibels. The offset has a range of +255 to −256 and is normalized to the range of +12 to −12 in this table.

Figures 12 and 13 show the variance in pixel intensity for all columns and rows, respectively, in the 256 × 256 region of the CCD FOV with gains and offsets for CCD outputs 1 and 2 of 8 dB, −5 and 10 dB, −5, respectively. As can be seen from these figures, the variation in pixel-to-pixel intensity within one column is smaller than the pixel-to-pixel variation within one row over the entire region. This result shows that equalization between the two outputs of the CCD is fairly consistent across the image area of the device. Comparing these figures additionally shows that the column plot exhibits a gradient, which swamps the column pixel variation. The gradient appears to be due to the camera and not to the light source. This observation comes from the fact that rotating the camera with respect to the integrating sphere does not affect this phenomenon in the pixel data.

B. ROI Results

ROI operation produces an additional artifact in the pixel data that is dependent on the ROI location. ROIs located in the upper-right quadrant of the CCD FOV have a fairly uniform flat-field response. ROIs in the lower-left corner exhibit a bright-band phenomenon that produces several very bright lines at the top of the ROI. This phenomenon is proportional to the number of line-scroll operations performed to

Table 7. Gain and offset mappings for a uniformly lit FOV.

Output 1 gain, dB	Output 2 gain, dB	Output 1 offset	Output 2 offset
6	8	−12	−7
8	10	−9	−6
12	13	−5	−5
13.8	15	—	—
18	20	—	—

reach the first line of a ROI and is exhibited by an increase in the number of bright lines and the intensity of those lines of pixel data compared to subsequent lines in the ROI. This effect is mitigated for a given number of line-scroll operations as the ROI moves toward the right-hand side of the CCD FOV.

In order to assess this effect in a realistic environment, a mapping of the gains and offsets for dual-line output in the Optical Communication Demonstrator (OCD) platform was defined as given in Table 8. The OCD platform uses a telescope assembly for the camera optics to focus a collimated laser beam down to a spot on the CCD image area. Functionally, this arrangement provides a black background with a single bright spot within the CCD FOV as the target signal during a tracking experiment. This mapping is somewhat different from the mapping for the flat-field case discussed previously. It most likely stems from the difference in pixel amplitude within the TLV987s due to the use of a black background during line-scroll operations versus a uniformly lit background, discussed in Section VII.A.

The bright-band effect can be seen in Fig. 14 for a ROI with its origin located at 400,400 and having an extent of 40 pixels in both axes. In this figure, the gains and offsets of the two TLV987s are adjusted per the mappings in Table 8 to set the spot peak intensity to approximately half of the TLV987 dynamic range. As can be seen from this figure, the ROI shows a very pronounced bright band in the first 10 rows of pixels and requires increasing of the vertical dimension of the ROI by this many rows to avoid spatial overlap of the target spot with the band.

This bright-band characteristic appears to be emanating from within the TLV987 signal processors. The primary reason for this conclusion is as follows. The camera was run using the same flat-field image and operational parameters for the CCD imager chip but using different gains and offsets for the signal processors. It was observed that an increase in the pixel intensity (at the output of the signal processors) resulted in an increase in both intensity and broadening of the band. The analog pixel data into the signal processors were not changed during this test. This effect was observed by increasing the gain and/or the offset in the TLV987s. Thus, the pixel amplitude after the offset and gain stages in the signal processors correlates with this phenomenon but the amplitude of the analog pixel data into these devices does not.

An additional reason for citing the signal processors for this effect comes from the change in the size of the bright band when using single-line versus dual-line readout from the CCD. For this test case, the image and operational parameters for the CCD were held constant as were the operational parameters of the signal processors. Images were taken using either single- or dual-line output from the CCD. The observed result is that the dual-line case resulted in twice as many bright rows of pixels as compared with the single-line case. This behavior is consistent with a charge buildup problem in the signal processors. Clearing the charge in the node after the offset and gain stages of the TLV987s requires a certain number of pixel read operations to reach a steady-state condition. The end result is twice as many bright rows to clear both TLV987s. Validation of this conclusion was confirmed by replacing the TLV987s with TLV9907 signal processors in a Revision 2 build of the TC237 CCD card and confirming the absence of this phenomenon.

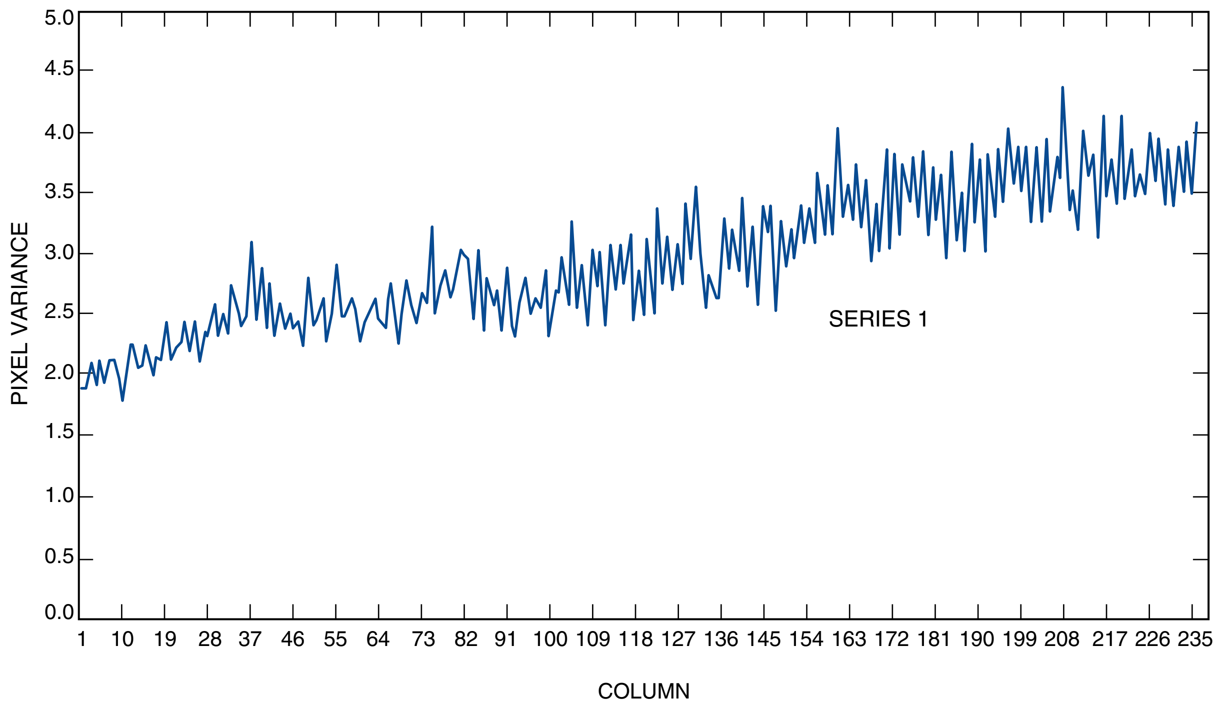

Fig. 12. Column variance (8, –5, 10, –5).

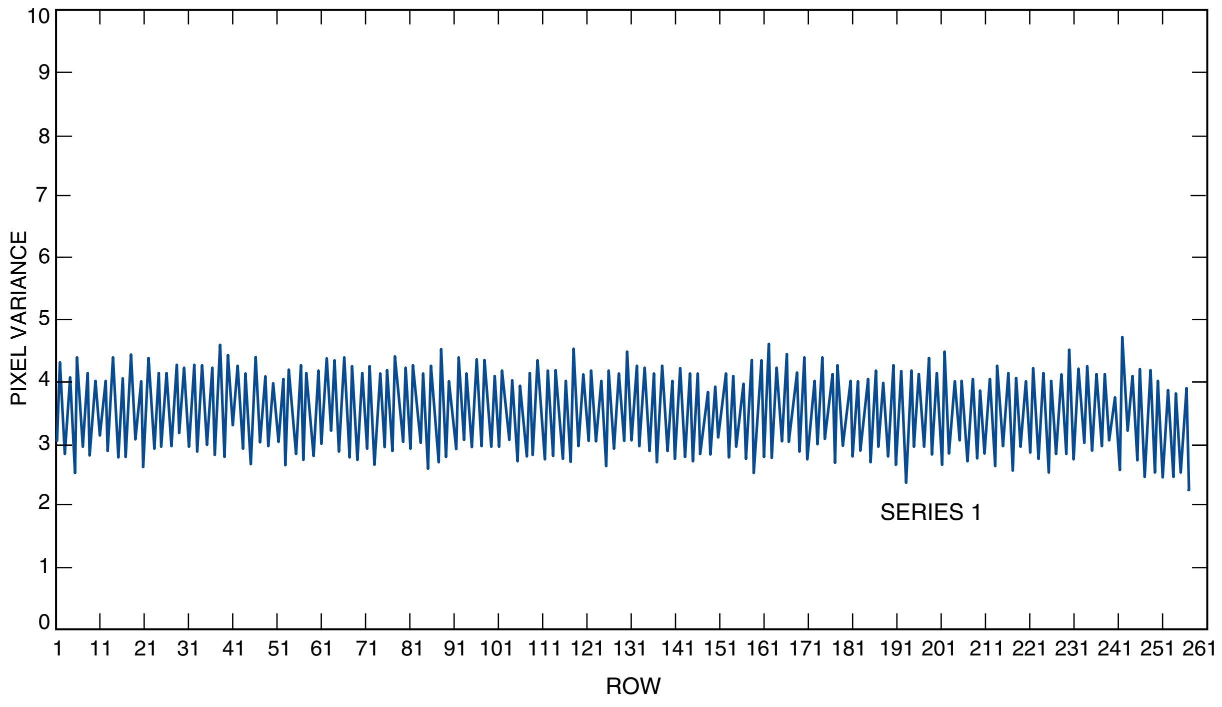

Fig. 13. Row variance (8, –5, 10, –5).

Table 8. Gain and offset mappings for the OCD platform.

Output 1 gain, dB	Output 2 gain, dB	Output 1 offset	Output 2 offset
0	0	−12	−10.6
35	35	−9	−6.4

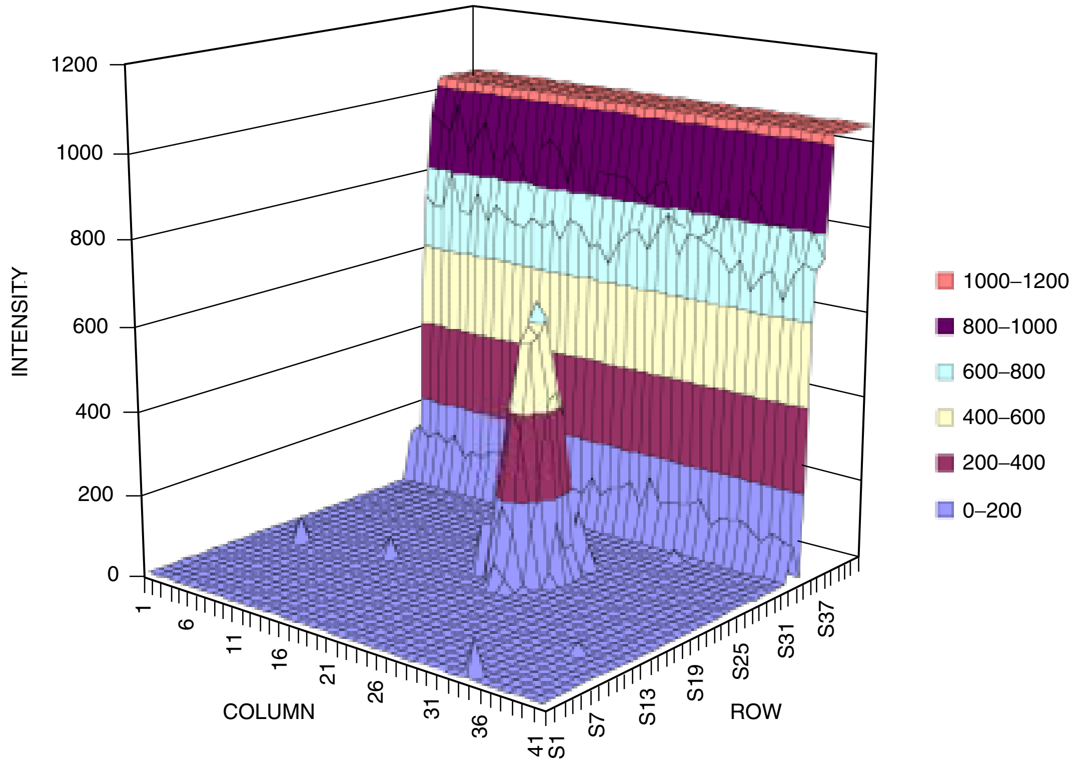

Fig. 14. ROI bright-band effect with ROI origin and extent of 400,400 and 40× 40, respectively.

VIII. Conclusions

The goal of the RARE camera development is to provide a key component for a real-time, adaptive tracking platform. We have developed this infrastructure by implementing a methodology to quickly extract pertinent pixel data using a commercially available progressive scan imager. This technology is also well-suited to adjusting the camera parameters to accommodate changing ambient and target conditions during tracking.

In this article, we presented details of the RARE camera design based on the Texas Instruments TC237 CCD imager chip. The novel feature of this design is the use of an event-driven paradigm for imager control. This capability was implemented by developing a custom FPGA controller that converts a commercially available CCD imager into a smart pixel device.

Communication to the FPGA controller is via commands from a host tracking processor. This combination of FPGA controller and host tracking processor provides for higher-level commands to handle low-level imager operation for dynamically controlled ROI capability on a per-frame and intra-frame basis. The result is the Version 2.0 RARE camera design with the features listed in Table 1. Additional features are envisioned for the Version 3.0 design, as shown in Table 1.

To assess the speed performance of the Version 2.0 designs, several experiments were conducted for single- and dual-ROI operation. The first experiment illustrated the change in frame rate for a single fixed ROI origin and varying ROI size. The second experiment illustrated the change in frame rate for a single fixed-sized ROI with varying origin from the first row (top) and first column (left) of the CCD FOV. The results from these experiments achieved 75 percent of the theoretical best-possible frame rates for this CCD imager in the configuration detailed in Section VI.B. The difference between the theoretical limit and the experimental results is due to a combination of both fixed and variable delays in the FPGA logic and host tracking processor software, as discussed in Section VI.B. A third set of experiments was conducted to illustrate the change in frame rate from two counter-propagating ROIs, as detailed in Section VI.C.

The frame rate versus single-ROI position or size characterized by the curves in Figs. 7 and 8 is to be expected for a camera using a progressive scan imager. For a 10×10 ROI, the maximum achieved frame rate was 6.4 kHz with the origin located at (0,0), and the minimum frame rate was 1.1 kHz with the origin at (400,400). The minimum frame rate for a given ROI size places the upper limit on the CCD frame rate and system update rate. This rate can be increased by limiting the sub-area of the FOV that the ROI is allowed to occupy, with the minimum rate for a given-sized ROI shown in Fig. 8.

The dual-ROI results presented in Tables 3 through 6 (and Figs. 9 through 11) show that frame rates of 900 to 1100 Hz are achievable for two 11×12 ROIs centered about the CCD FOV. The two ROIs move in a counter-propagating fashion to emulate the operational mode of the envisioned acquisition and tracking platform. These frame rates will vary if the ROIs are allowed to move in a co-propagating fashion, as would be the case for the tracking of two targets within a scene. The frame rates for this case are governed by the ROI sizes and locations shown in Figs. 7 and 8 and by the relative positions of the ROIs as shown in Tables 3 through 6 (and Figs. 9 through 11).

The dual-line ROI results presented in Section VII show that 10 percent uniformity between CCD pixel outputs is possible with proper scaling of the gain and offset parameters of the TLV987 signal processors. The scaling is facilitated by a priori selection of the offset parameters and then determination of the high- and low-gain settings for equalized operation. Conversely, the high- and low-offset settings can be determined for fixed-gain settings. Once defined, the mapping can be used to dynamically adjust the two TLV987 signal processors to compensate for scene dynamics with equalized pixel amplitude to less than 10 percent disparity between the pixel-stream amplitudes.

The primary issue with ROI operation is a bright-band artifact that is most pronounced for ROIs in the lower-left quadrant of the CCD FOV. This artifact is due to charge build up in the TLV987 processors during line-scroll operations. The severity of the charge build-up is proportional to the signal amplitude after the gain stage in the 987 chips and imposes a penalty of increasing the vertical size of the ROI to remove this charge build-up. Substitution of the TLV990 [7] in place of the TLV987 for the Revision 2 CCD card eliminated this effect.

A nearly two times speedup is possible by running the imager at its maximum possible clock speed of 20 MHz per pixel stream. Additional speed increases require using a different progressive scan imager with more than two pixel outputs to provide more parallelism in accessing the image data. Due to the generality of the RARE camera control scheme, the FPGA controller can be used for other commercially available CCD imagers to optimize system performance in terms of speed, image quality, or other parameters of interest.

Acknowledgment

This is a derivative article based on a paper published by the SPIE—The International Society for Optical Engineering: S. Monacos, R. Lam, A. Portillo, and G. G. Ortiz, "Design of an Event-Driven Random-Access-Windowing CCD-Based Camera," presented paper, *Proceedings of the SPIE Conference in Free-Space Laser Communications Technologies XV*, San Jose, California, January 2003.

References

[1] S. P. Monacos, A. A. Portillo, W. Liu, J. W. Alexander, and G. G. Ortiz, "A High Frame Rate CCD Camera with Region-of-Interest Capability," *2001 IEEE Aerospace Conference Proceedings,* Big Sky, Montana, March 10–17, 2001.

[2] A. Talukder, J.-M. Morookian, S. Monacos, R. Lam, and C. Labaw, "Real-Time Non-Intrusive Eyetracking and Gaze-Point Determination for Human-Computer Interaction and Biomedicine," 2nd WSEAS International Conference on Signal, Speech and Image Processing (WSEAS ICOSSIP 2002), September 25–28, 2002.

[3] Texas Instruments, *680x500 Pixel CCD Image Sensor TC237,* June 1996.

[4] Texas Instruments, *3-V 10-Bit 27 MSPS Area CCD Sensor Signal Processor TLV987,* September 1999.

[5] QuickLogic Corporation, *QL5064 User's Manual,* Rev. A, December 1999.

[6] Transtech Corporation, *PMC-FPGA01 User Manual,* 2001.

[7] Texas Instruments, *TLV990-28 3-V, 10-BIT, 28-MSPS Area CCD Analog Front End,* August 2000.