A 3.8ms latency correlation tracker for active mirror control based on a reconfigurable interface to a standard workstation

Mark Shand*

Digital Equipment Corporation, System Research Center
130 Lytton Av., Palo Alto, CA 94301, USA

Wang Wei

Swedish Royal Academy of Sciences, Research Station for Astrophysics
Stockholm Observatory, S-133 36 Saltsjöbaden, Sweden

Göran B. Scharmer

Swedish Royal Academy of Sciences, Research Station for Astrophysics
Stockholm Observatory, S-133 36 Saltsjöbaden, Sweden

ABSTRACT

We describe the use of a reconfigurable interface board based on FPGAs and a UNIX workstation to implement a correlation tracker with 3.8ms latency. The correlation tracker is part of an active mirror system in use at the Swedish Vacuum Solar Telescope, La Palma, Canary Islands.

The reconfigurable interface is used to leverage the workstation CPU, relieving it of tasks that it performs poorly such as rapid context switching and low-level bit manipulation. The reconfigurable interface handles control of external devices, high-performance input (16 MB/s) and data preformatting. The workstation CPU, a 64-bit microprocessor, performs the bulk of the computation. For the key computations of the correlation tracker we are able to treat 8 pixels in parallel in the CPU's 64-bit integer datapath.

We present the structure of the CCD interface configuration and the implementations of the key algorithms on the workstation CPU. We describe the design trade-offs that arose during the development of the system, and demonstrate the symbiosis between components implemented in software and configurable hardware.

keywords: adaptive optics, solar astronomy, cross-correlation, reconfigurable hardware, FPGA, real-time, control systems, UNIX, workstation, hardware-software co-design.

*The work described in this paper was carried out at Digital Equipment Corporation's Paris Research Laboratory and at the Research Station for Astrophysics of the Royal Swedish Academy of Sciences. The authors' e-mail addresses are shand@acm.org, wang@astro.su.se, scharmer@astro.su.se

1 INTRODUCTION

Microprocessors found in today's workstations are powerful compute engines, currently capable of hundreds of millions of instructions per second, and growing more powerful every year. Their programmability makes it easy to apply this processing power to a wide variety of tasks. However, for real-time tasks that require guaranteed response to external events and customized external interfaces some dedicated hardware front-end is needed.

Reconfigurable technology makes it possible to build a generic hardware front-end that can be customized, via programming, achieving flexibility similar to that of the microprocessor itself. The result is a powerful, adaptable, cost-effective system capable of high I/O bandwidth and real-time response that can be targeted to a variety of applications, and can easily take advantage of new generations of microprocessors.

Such systems, based on off-the-shelf UNIX workstations and modest sized reconfigurable interface boards, are used in almost all aspects of the image acquisition system at the Swedish Vacuum Solar Telescope (SVST), a solar observatory in La Palma, Canary Islands.

The most computationally challenging aspect of the SVST system is its active mirror control system, which is the subject of this paper. Through its use of programmable components, the system is able to incorporate the following innovations:

- Ability to switch between quad-cell and cross-correlation modes for image tracking.
- Support of a 32×32 live image and 40×40 reference image in cross-correlation mode.
- Ability to reduce magnification by subsampling during poor observing conditions.
- Use of a cross-correlation controlled tilting mirror system for routine solar observing.
- Ability to scale performance over time through the use of standardized commodity components.

The discussion is arranged as follows. Section 2 gives an overview of the SVST image acquisition system. Section 3 discusses the reconfigurable interface board and its use at SVST. Section 4 discusses the implementation of the active mirror control system. Section 5 presents results and status.

2 OVERVIEW OF THE SVST IMAGE ACQUISITION SYSTEM

To answer many current questions in solar physics, astronomers must gather and analyze high resolution images and spectra of localized regions of the solar surface. In many cases images are needed every few seconds so that sequences stretching over hours can be built up. These sequences are used to study the evolution of solar structures.

Atmospheric turbulence creates a phenomenon astronomers refer to as *seeing* which hinders the gathering high resolution solar images. The dominating effect of seeing is rapid image motion which causes image blurring if the exposure time is longer than approximately 20 ms and equally rapid changes in image sharpness.

Figure 1 presents an overview of the image acquisition systems in use at SVST.[1] Each CCD image sensor is interfaced to a workstation through the reconfigurable interface described in section 3. All images are transferred directly to workstation memory. Images from the observational CCDs are treated on-line by the workstation's general-purpose microprocessor using techniques to mitigate the effects of seeing such as frame selection and averaging. These techniques are well-established, though in the past it was not possible to realize them on-line in

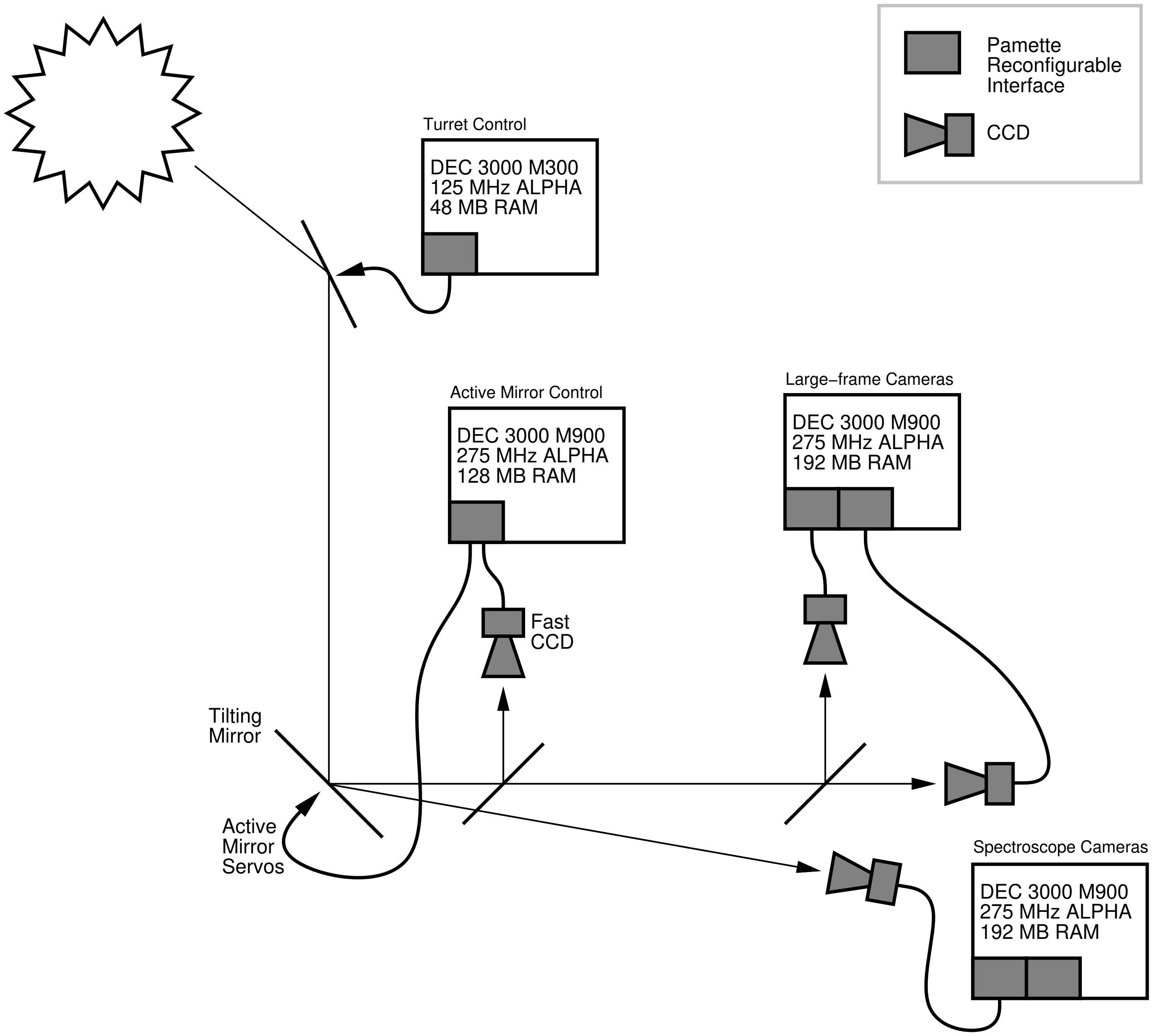

Figure 1: SVST image acquisition system

software. Several observational CCDs may be used in parallel in support of advanced off-line wavefront sensing and reconstruction techniques.[2]

2.1 Role of active mirror control

The effects of seeing may be further reduced by sensing the atmospherically induced distortions in the incoming wavefront and applying compensating distortions to the telescope optics. The SVST system monitors incoming images with a rapid frame CCD. The reconfigurable interface formats the incoming images and delivers them efficiently to the workstation memory. Software on the workstation computes the image motion in real-time and applies a compensating tilt to a flat mirror in the path to the observational CCDs to reduce motion induced blur in prolonged exposures. This control system must operate several times faster than the image motions it compensates which at the La Palma site have been measured to be largely below 40Hz.

Image motion is computed by one of two techniques:

Quad-cell mode divides the image or part of the image into 4 quadrants, the integrated intensities of which are used to keep a roughly circular feature such as a sunspot or pore centered in the frame.

Cross correlation mode calculates the cross correlation of the image against a reference image for some limited set of displacements. This technique can lock onto any arbitrary feature of the solar surface such as irregular pores or granulation.

Quad-cell tracking is preferred when features in the target image permit its use. However such features are not always present, particularly in the coming years due to the current trough in the sunspot cycle. In such cases cross-correlation tracking must be used.

As more processing power becomes available the flat mirror will be replaced by a segmented or deformable mirror leading to a fully fledged adaptive optics system. The current software algorithms and reconfigurable interface will be directly applicable to this new more capable system.

3 THE PAMETTE RECONFIGURABLE INTERFACE BOARD

Figure 2 shows the Pamette reconfigurable interface board for TURBOchannel[3] developed by members of the PAM Project[4] at Digital Equipment Corporation's Paris Research Laboratory. TURBOchannel is a simple but efficient I/O bus used in Digital workstations prior to the advent of PCI. It supports reads and writes from the CPU and DMA between I/O options and the memory system. Since 1989, TURBOchannel equipped workstations have been available supporting DMA throughput in excess of 90MB/s.

As can be seen from the figure, the Pamette board is very Spartan. It contains a handful of interface chips and clock generators, a PROMed Xilinx XC3142-3 for bootstrap, a user-programmable area consisting of four footprints suitable for any Xilinx 4000 series device in the PQ208 package[5] and extension connectors. In volume production the cost of Pamette is dominated by the FPGAs. Figure 2 shows a Pamette populated with four XC4010 devices. Another configuration, popular at SVST because of its reduced cost, has one XC4010, one XC4003H and two empty positions. Extension connectors allow direct external connection as in figure 2 (right side), or connection to a daughter board, as in figure 3, which shows a differential to single-ended converter board, suitable for connection to a variety of CCD cameras.

The Pamette philosophy is that the entire digital part of an interface can be implemented in the user-

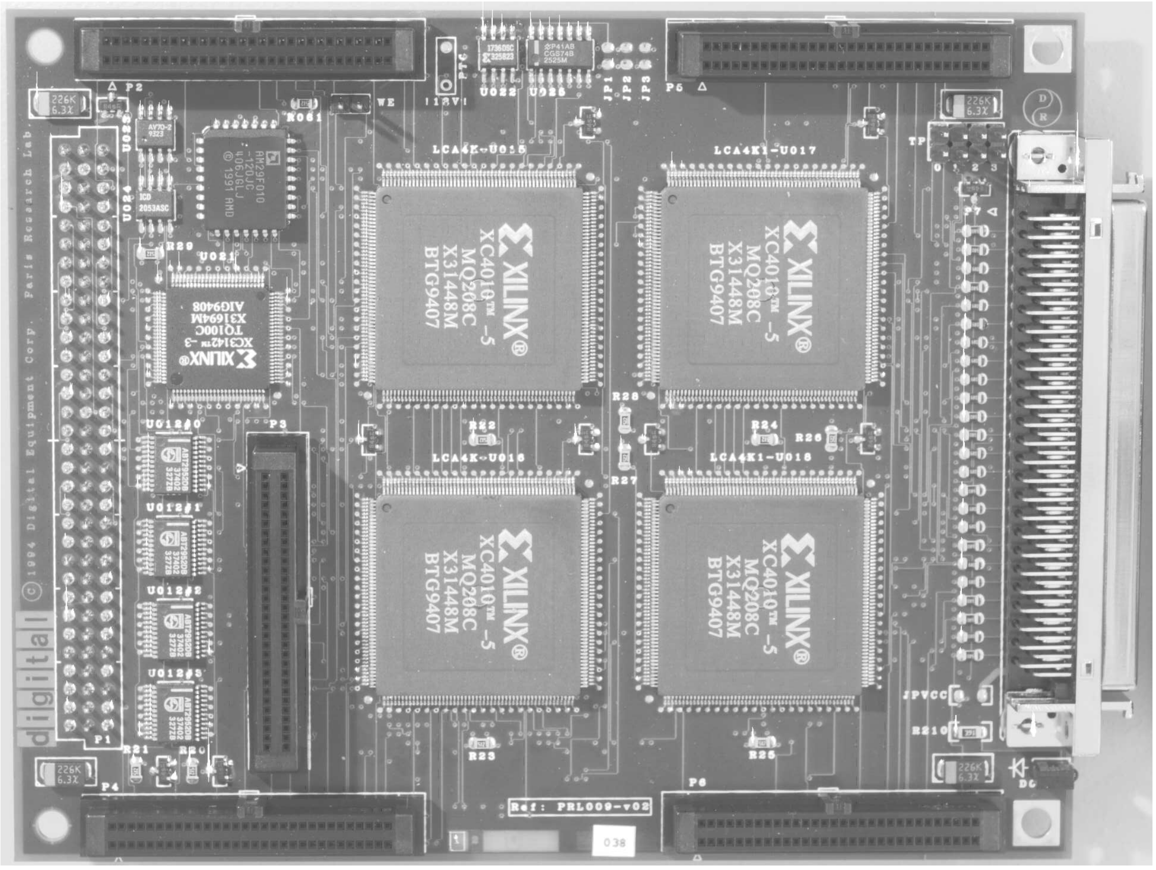

Figure 2: Pamette reconfigurable interface board for TURBOchannel (actual size)

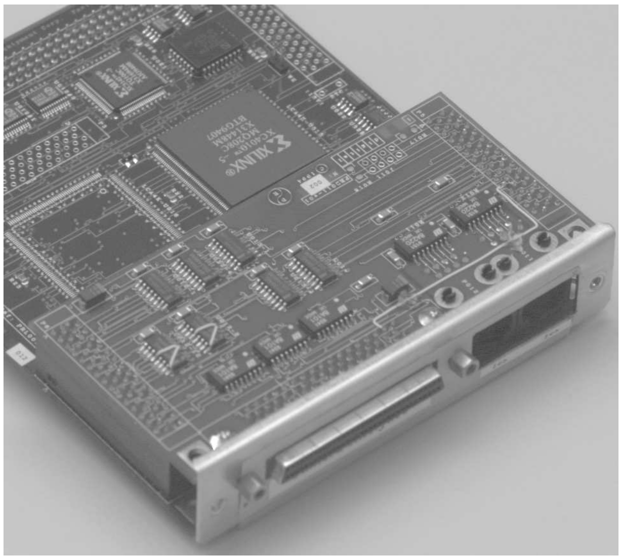

Figure 3: Pamette board with differential receiver daughter board

programmable Xilinx 4000 series devices. Daughter boards, which are cheaply and easily built, support differing physical connectors and electrical signalling conventions.

4 ACTIVE MIRROR CONTROL SYSTEM IMPLEMENTATION

4.1 Dalsa CCD camera

A Dalsa CA-D1 128 × 128 CCD with 16μm pixels is used to monitor incoming images. The CCD is operated in free running mode so the integration time of each image is equal to the readout time. The pixel readout rate is 16MHz yielding whole images at 840Hz. Note that this is well above the maximum rate of change of seeing. The pixels are sampled to 8-bit resolution and presented 8-bit parallel on differential pairs to the interface board.

4.2 Pamette configuration for CCD interface

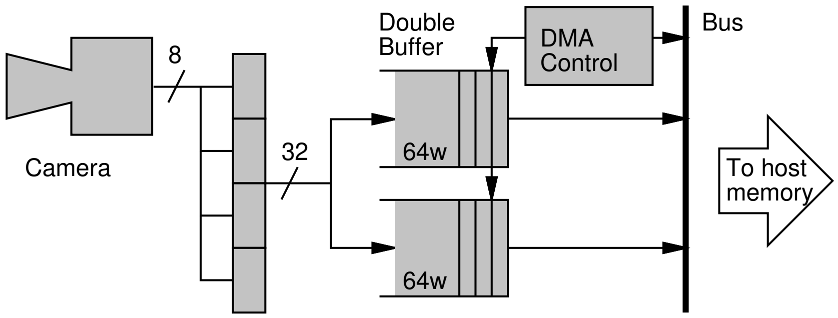

Figure 4: Pamette configuration for CCD interface

The main function of the CCD interface is to take the steady stream of pixels produced by the CCD, pack them into blocks of 32-bit words and transfer the blocks in bursts via DMA to an application on the host, a Digital Equipment Corporation DEC 3000 M900 Alpha workstation[6] running Digital UNIX. A combination of interface configuration and application software together implement this function.

Block buffering is performed in the FPGAs using the Xilinx 4000 series RAM capability. DMA latency measurements made using the interface itself[7] show that a few hundred bytes of buffering is adequate for our needs. Buffers are organized in a double-buffer scheme so that one buffer can receive incoming pixels while the other is requesting DMA (always of a full buffer). At 16MB/s input rate we cannot afford to copy DMA buffers. Blocks are transferred directly to the application image buffer. This buffer is allocated through normal process memory allocation techniques and consists of physically discontiguous pages. TURBOchannel DMA however is physically addressed*. A page is filled every 600 μs. To reduce interrupt load and avoid a requirement for rendezvous with software at the transitions between pages, we use the following multi-level DMA scheme. After allocating the image buffer, the application builds a list of command blocks (illustrated in figure 5) consisting of flags, length, buffer physical address and optionally a pointer to the next command block (for use when it is not physically contiguous to the current command block). The application enables DMA by loading the physical address of a command block to the configured Pamette. The Pamette then repeatedly loads a command block through a DMA read and performs DMA writes of image data until the region encoded in the command block

*Newer Alpha-based workstations equipped with TURBOchannel alleviate the problems of physically addressed DMA through the provision of hardware scatter/gather maps but these are not adequate for our application which may allocate almost all of system memory to image buffers.

is exhausted. The command blocks can be viewed as an instruction stream that controls execution of the DMA engine.

length	flags
image destination address	
next command address	

Figure 5: DMA Command Block

The CCD interface continuously fills a ring buffer several images deep. Implementation is by a command block list that loops back on itself after some number of frames. Since the CCD produces an infinite stream of images the DMA never stops. Nevertheless software must maintain some kind of synchrony with the DMA engine to ensure it is processing recent data. One of the command block flag bits allows an interrupt to be raised when that command block is fetched. Thus it is possible to alert host software at any point within an image transfer. We exploit this to minimize latency in our two operating modes:

Quad-cell mode performs a relatively rapid computation on the entire image. An interrupt is raised at the end of each frame.

Cross correlation mode is computationally expensive and can therefore only be run on a subwindow. By choosing the subwindow at the top of the frame and raising an interrupt as soon as this part of the image is transferred we can partially offset the higher latency introduced by this inherently slower computation.

Although Digital UNIX provides process scheduling features in support of real-time tasks, we have not found these to be necessary for synchronization with the CCD interface. Instead, on receipt of an interrupt the software polls internal address and status registers in the DMA engine to determine how far it has progressed since it raised the interrupt. In the rare occasions where the process has been preempted for a significant period after the interrupt was raised, software can elect to drop that frame and skip to the frame following.

4.3 Software techniques for fast image processing

The distinguishing feature of the SVST active mirror control system is its reliance on software implementations of image processing algorithms on a general purpose microprocessor. Use of software rather than hardware has numerous advantages in terms of maintainability, ease of experimentation and future upgradability. However, with current microprocessors, careful implementation is required to achieve the necessary performance, particularly in the case of cross-correlation.

Rather than treating byte-wide pixels individually, wherever possible we pack eight pixels into a 64-bit word and treat them in parallel in the 64-bit wide Alpha CPU integer datapath. Cross-correlation compares various translations of an incoming image against a reference image to determine the translation that yields the closest possible match. Ideally the comparison uses a sum of squares of differences. Instead, to avoid expensive multiplications we use a sum of absolute differences. Computing absolute values normally uses conditional operations on each data element however a straightline sequence of unconditional boolean and arithmetic instructions can compute the same function. Applying this unconditional sequence to several data elements in parallel more than offsets its extra cost in basic operations. For the SVST cross-correlation implementation the input pixels are scaled to seven bit quantities to accommodate a guard bit on each pixel. The resulting algorithm is illustrated in pseudocode in figure 6.

```
// V is an 8 element vector held in one machine word
// v[i] is the ith element of V

// R = reference image
// B = new image
// R = r[7], r[6], .. r[0]        0 <= r[i] < 128
// B = b[7], b[6], .. b[0]        0 <= b[i] < 128
// R is preprocessed to be biased by 128
// A = a[7], a[6], .. a[0]        a[i] = r[i]+128, 128 <= a[i] < 256

C = A - B                      // c[i] =128+r[i]-b[i], 0 < c[i] < 256
H = C & 0x8080808080808080     // extract top bits of bytes
X = H ^ 0x8080808080808080     // x[i] = 128 IFF r[i]-b[i] < 0
L = X >> 7                     // l[i] = 1 IFF r[i]-b[i] < 0
X = H | X-L                    // x[i] = 127 if r[i]-b[i] < 0, else 128
C = C ^ X                      // if r[i]-b[i] < 0, one's compl. else
                               // subtract 128
C += L                         // add one to one's complemented bytes
                               // to form two's complement.
                               // Now c[i] = abs(r[i]-b[i])
C += C >> 8                    // add odd bytes to even bytes
                               // we can guarantee no overflow since
                               // abs(r[i]-b[i]) < 128
C &= 0x00ff00ff00ff00ff        // clear odd bytes, this accommodates
                               // overflow on subsequent adds
C += C >> 16                   // add c[0] += c[2], c[4] += c[6]
C += C >> 32                   // add c[0] += c[4]
                               // Now c[0] = sum(abs(a[i]-b[i])) : i=0..7
```

Figure 6: 8 byte parallel sum of absolute differences

4.4 Servo-control

The servo-system that positions the active mirror needs several low-bandwidth parallel connections to the controlling workstation. These are provided by registers in the Pamette board that can be read, or written, from the host and whose inputs, or outputs, are connected to otherwise unused external connectors. The latency from CPU to servo-system through this path is a fraction of a microsecond.

5 RESULTS AND STATUS

Figure 7 summarizes the performance of the operating modes of the SVST active mirror control system. The current implementation of the cross-correlation algorithm requires more than one frame time of computation so this mode only uses every second frame from the CCD. Since the steps are pipelined, the number of corrections per second is determined by the slowest step rounded up to an integral number of frames. Latency on the other hand is the sum of all the steps.

In cross-correlation mode the system interpolates between the best correlation matches and has proved capable

Operation	Quad-cell mode 128 × 128	Cross-correlation mode live: 32 × 32, ref: 40 × 40
CCD exposure time	1.2 ms	1.2 ms
Read-out time	1.2 ms	0.6 ms
Computation time	1.0 ms	2.0 ms
Corrections/second	840	420
Total latency	3.4 ms	3.8 ms

Figure 7: Tracker latencies

of resolution to 0.01 pixels. Figure 8 shows the response of the system viewing an simulated target when subjected to an impulse produced by touching the laboratory bench on which it was placed. The vertical axis is the measured displacement in pixels, the horizontal axis is the frame number.

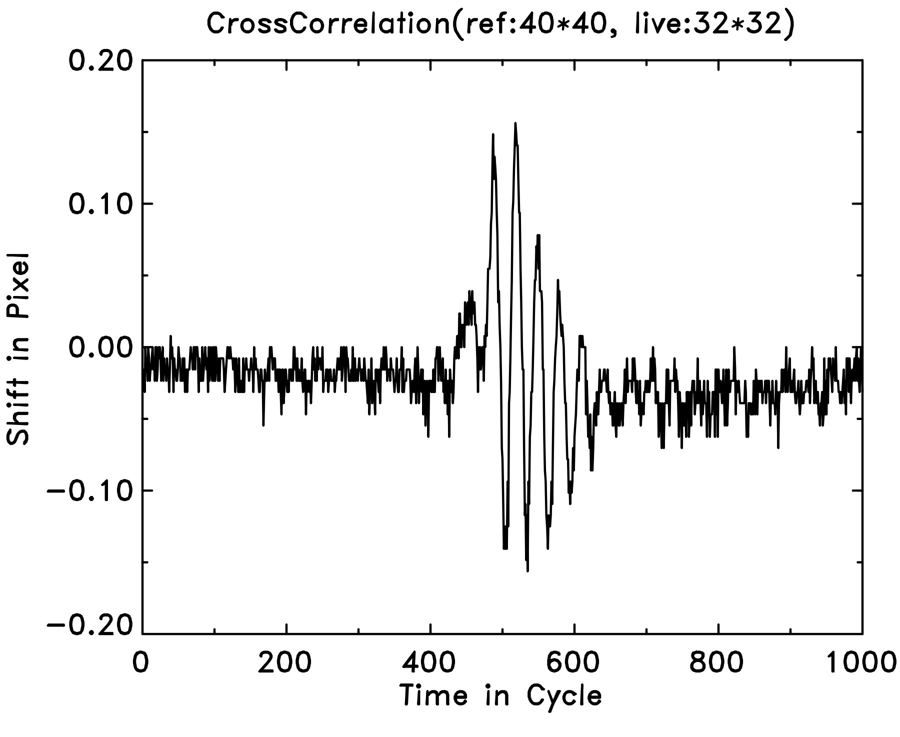

Figure 8: Cross-correlation tracker operation

The system first became operational in August 1994. It was enhanced at the end of 1994 with a new interface configuration, faster microprocessor and a refined interface/software synchronization mechanism. It has been in routine use throughout the 1995 observing season. The performance of the current system is limited by the servo-motors on the active mirror whose response is limited to 30Hz. Despite this weakness the system has proved extremely reliable. We are actively exploring new servo systems that will improve this performance, and ultimately enable to a segmented or deformable mirror system.

6 CONCLUSIONS

A combination of reconfigurable interface and software running on a standard workstation has proved capable of implementing a robust correlation tracker suitable for routine use in solar observation. The correlation tracker currently supports a window size of 32×32—larger than the fixed window size of many hardware correlation trackers.

By basing all key elements on programmable components, the same system implements quad-cell tracking in addition to correlation tracking.

Development time is dramatically faster and costs, much cheaper, than building specialized systems for these tasks. Maintenance cost are also greatly reduced because a maximum of standard hardware and software is used.

The use of standard workstation technology allows easy performance upgrades of the operational system, by taking advantage of hardware and software compatibility of successive workstations generations. Normally this is not the case with specialized systems which cannot easily exploit advances in technology.

7 ACKNOWLEDGEMENTS

The Pamette reconfigurable interface board, its CAD system, firmware and runtime support are products of the PAM project which was active at Digital Equipment Corporation's Paris Research Laboratory from 1988 to 1994. The author thanks the other members of the PAM project: Patrice Bertin, Philippe Boucard, Didier Roncin, Hervé Touati and Jean Vuillemin.

Digital Equipment Corporation and the Swedish Royal Academy of Sciences provided financial support for this work.

8 REFERENCES

[1] M. Shand, *Flexible Image Acquisition using Reconfigurable Hardware*, IEEE Workshop on FPGAs for Custom Computing Machines, April 19-21 1995.

[2] M. G. Löfdahl, G. B. Scharmer, *Wavefront sensing and image restoration from focused and defocused solar images*, Astronomy and Astrophysics Supplement Series, October 1994.

[3] Digital Equipment Corporation, *TURBOchannel Specification Revision C*, DEC document EK-TCDEV-DK.

[4] J. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, P. Boucard, *Programmable Active Memories: Reconfigurable Systems Come of Age*, IEEE Transactions on VLSI Systems, to appear 1995.

[5] Xilinx, Inc., *The Programmable Gate Array Data Book*, Xilinx, 2100 Logic Drive, San Jose, CA 95124, USA, 1994.

[6] Digital Equipment Corporation, *DEC 3000 Model 800/800S/900 AXP Information Kit*, DEC Document EK-FLMWS-DK.

[7] M. Shand, *Measuring system Performance with Reprogrammable Hardware*, PRL report 19, Digital Equipment Corp., Paris Research Laboratory, August 1992. *ftp://gatekeeper.dec.com/pub/DEC/PRL/research-reports/PRL-RR-19.ps.Z*