Real Time Processor for Array Speckle Interferometry

Gordon Chin
Planetary Systems Branch, Code 693

Jose Florez, Renan Borelli, Wai Fong,
Joseph Miko, and Carlos Trujillo

Instrument Electronic System Branch, Code 728

NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Abstract

With the construction of several new large aperture telescopes and the development of large format array detectors in the near IR, the ability to obtain diffraction limited seeing via IR array speckle interferometry offers a powerful tool. We are constructing a real-time processor to acquire image frames, perform array flat-fielding, execute a 64 x 64 element 2D complex FFT, and to average the power spectrum all within the 25 msec coherence time for speckles at near IR wavelength. The processor will be a compact unit controlled by a PC with real time display and data storage capability. In this manner, we will provide the ability to optimize observations and obtain results on the telescope rather than waiting several weeks before the data can be analyzed and viewed with off-line methods.

Introduction

Observations on large ground-based telescopes still have important roles with the advent of the Hubble Space Telescope (HST) and other planned space-based observatories such as the Space Infrared Telescope Facility (SIRTF). One role will be to use the full aperture of large ground-based telescopes to resolve objects which HST or SIRTF cannot do. HST is a 2.4 meter telescope with a diffraction limit in the visual of 70 mas (1 mas = 10^{-3} arcsec). SIRTF will have a fairly small telescope, 80 cm, and at 2 to 10 μ m wavelength will have diffraction limited resolution of a few arcsecs. The Keck 10-meter telescope on Mauna Kea, when completed, will allow both visual and infrared observations (Jones [1]) and will have a visual resolution of 17 mas (at 2 μ m it will be about 70 mas). However, ground-based telescopes are limited by atmospheric seeing to about 1 arcsec or less on good nights.

Ground-based observing is limited by speckling. Atmospheric turbulence cells over the aperture of the telescope cause random phase differences and phase tilts of the observed object resulting in speckles. Speckling can be seen as the twinkling of stars. Speckle interferometry is a technique which can remove distortion from the atmosphere by taking images during the coherence time of the speckles, i.e. during the period when the atmosphere does not change the speckles. By computing a two dimensional Fourier transform of the speckle images and averaging the power spectrum, the high spatial frequency information up to the diffraction limit can be retained while random variations due to the atmosphere is eliminated (Labeyrie [2]).

Since only the power spectrum is obtained with speckle interferometry, phase information from the observed object is lost. The result is the auto-correlation of the object. The original image can often be deduced if the object is very simple such as a pair of point sources. However for more complex images, an image restoration algorithm (the Knox-Thompson technique [3]) can produce a speckle-derived image if phase differences between pixels are also calculated.

We can compare the speckle technique with classical Michelson interferometers. Speckle interferometry uses the entire aperture of the telescope and therefore can observe fainter objects than a classical Michelson interferometer whose optics are limited by atmospheric seeing to 25 cm. On the other hand, the baselines for a Michelson interferometer may be much longer than single dish apertures and thereby can attain higher resolution than speckle interferometry.

Speckle interferometry has been applied to measuring stellar diameters and resolving double stars (e.g. Gezari, Labeyrie, and Stachnik [4]), to image the Pluto-Charon system and (e.g. Baier and Weigelt [5]), and to the recent discovery of a bright companion to the supernova SN 1987A (Nisenson *et al.* [6]; Meikle *et al.* [7]).

Image Acquisition and Processing

Speckle interferometry requires enormous amounts of information to be stored and processed. This is due to the fact that the coherence time for speckles ranges from about 5 msec for visual wavelengths to about 25 msec for the near infrared (IR) spectral range. Therefore it is necessary to store 40-200 images per sec.

Speckle images were originally stored on photographic film and the Fourier transform was then done by optical means [4]. However, due to the limited dynamic range of film emulsions, detector arrays or other types of electronic imaging devices have replaced films. Data storage and processing are currently handled exclusively by digital techniques (Nisenson *et al.* [8]).

For moderate array sizes, the amount of stored data corresponds to several hundred MBytes of information for one hour integration on a single object. Typical experiments include observations of several sources as well as several references. This information must then be processed off-line; that is 40-200 two dimensional (2D) complex fast Fourier transforms (FFT's) must be performed and averaged for every sec of real-time observing. If the Knox-Thompson image restoration algorithm is performed, further calculations are needed.

Usually a delay of several weeks or months occur before the data can be fully processed. This delay seriously affects the observing process and eliminates the possibility of optimizing observations on the telescope. For example, the discovery of the bright companion to SN 1987A underlines this point [6]. The bright companion to SN 1987A was discovered by accident. The original intent of the experiment was to measure the diameter of the supernova shell. When further observations were performed several months later, the new observations could not detect the companion and now the nature of the original discovery is disputed (Burke [9]).

Large and sensitive near IR detector arrays are currently in use at several astronomical facilities (McLean [10]; Fowler *et al.* [11]; Forrest [12]). Several advantages accrue for speckle observations in the near IR. The first is the longer coherence time of the atmosphere as compared to the visual as mentioned above. The second is the fact that longer wavelength observations are more

advantageous in observing proto-stellar or proto-planetary systems in which the natal environment is highly dusty and therefore is highly obscuring in the visual. Christou *et al.* [13] discuss observing 2D near IR speckles with an array camera system.

It is now possible to construct a small and low cost processor to perform real-time speckle image processing. The processor can be constructed with off-the-shelf commercial digital components and signal processing chips. We will design the processor to acquire speckle images from a 64 x 64 pixel IR astronomical array camera, to perform the 2D complex FFT of the speckle images, and to average the power spectrum of the image in real-time (within 25 msec). In this manner we can eliminate the data storage burden associated with the speckle technique. At the same time we will be able to process the data and see the results instantly so that we can optimize the observing at the telescope.

In conjunction with this project, we are also participating in the design and fabrication of a chip set which utilizes a systolic architecture which will perform the FFT five times faster; a processing speed necessary to do visual speckle observations.

Design Criteria for a Real Time Speckle Processor

IR Detector arrays

Several factors dictate the design of our processor. The first is the need to interface to and to acquire data from a new type of IR detector array. These chips have a direct-read out (DRO) multiplexer scheme in which pixel elements are individually addressed and read rather than sequentially read out as in charge-coupled devices. Regions which are bright can be read out more frequently while faint areas can integrate longer to increase signal-to-noise. Non-destructive read-out of pixels is also possible.

The DRO arrays are of a hybrid design with specific detector materials such as InSb or Si:Ga bonded to the read-out electronics to achieve sensitivity in different spectral ranges. We want to construct a processor which can take advantage of the DRO multiplexer in order to observe with different arrays with the same electronics. The processor must be programmable in order to implement different read-out schemes and not to be fixed to a single detector type.

Data Acquisition Rate

The detector array must be read out within the speckle coherence time. Therefore the processor requires a large data throughput speed. We specify a maximum frame rate of 40 Hz for a 64 x 64 pixel array operating in the near IR. This rate is similar to that needed for observations at 10 μ m (thermal IR) where pixels must be read before background emissions from the telescope and atmosphere saturate the detector (Chin and Gezari [14]).

Flat Fielding

For speckle interferometry, flat-fielding is especially important. Residual structure which remains after flat-fielding will add erroneous spectral information to the data. The processor will need to interpolate between bad pixels, bad columns, and bad rows to remove array blemishes.

Intensity calibration is difficult. Pixel outputs become non-linear as they integrate due to changes in detector capacitance as the charge wells become full. Each pixel must have its own calibration curve to photometrically calibrate the detector array (Hoffman [15]). Therefore, detector calibration requires the evaluation of higher order polynomials or more complicated functions for each of the 4096 pixels (McCaughrean [16]).

2D FFT and Power Spectrum Averaging

We need 32-bit floating-point rather than fixed point calculations due to the large dynamic range in our observations. Intensity differences of 40-50 dB can be seen with the dynamic range possible on current IR detectors which have read-out noise of less than 100 e^- with a well depth of $10^6 e^-$ [16].

A 64 x 64 complex 2D FFT completed within 25 msec requires 20 million floating point operations per second (MFLOPS). Flat-fielding and power spectrum averaging both require an additional 10 MFLOPS. This computational power can only be achieved with a super computer (a Cray 1 does about 10 MFLOPS) or several array processors attached to a mini-computer.

A supercomputer is too costly and array processors are cumbersome and may not work for our application. Processor duty cycle is important due to the long integration times required for observations and the limited amount of telescope time available on telescopes. Array processors have a data transfer bottleneck getting information into and out of the mainframe's data buss. In addition, several array processors are required to work simultaneously in order to do data acquisition, flat fielding, power spectrum averaging, and the 2D complex FFT.

However, we can achieve the performance required for real time speckle processing with a compact and less costly solution.

Processor Architecture

Modular design with Common Hardware

The processor is based on a pipe-line architecture with modular units under micro-code control. Figure 1 is a block diagram of our system. The four independent units perform one of the required tasks; data acquisition, flat-fielding, 2D complex FFT, and power spectrum averaging. A personal computer such as an IBM AT with high resolution graphics display will provide the commands to the instrument and will be used to display and store the data. The PC is interfaced to the processor via an interface unit with 104-bit parallel board. The PC will be also be used to compile the micro-code instructions and to down-load the code to each unit.

We use as much common design in our units as possible. The data acquisition, flat-fielding, and the spectrum averaging units are essentially identical in design; the data acquisition unit has an integer rather than floating-point arithmetic chip. All units have identical micro-code control structures and interfaces with the PC. The 2D FFT module has a uniquely designed processing section devoted to the Fourier transform algorithm.

Pipe-line Data Flow

Data proceeds in a pipe-line fashion; all units operate in parallel on data obtained from their respective previous stages. The frame grabber interfaces to an array by providing the appropriate clock and address signals to read out the pixels within the coherence time of the speckles. The pixels are usually digitized to 14 to 16 bits of resolution. The array data is passed via ping-pong buffers to the flat-fielder which converts the array image to 32-bit floating point numbers and performs flat-fielding and intensity calibrations on each pixel. The flat-fielder will also maintain an average flat-fielded image accessible to the PC at the end of an integration period.

The flat fielded image is then passed to the 2D FFT unit which will perform the complex 2D Cooley-Tukey algorithm. The complex results go next to the accumulator unit via another set of ping-pong buffers. The power spectrum as well as its averages will be calculated in this unit and the results made accessible to the PC at the end of an integration period.

Frame Rate and Integration Period

Each unit will perform its functions synchronized to a frame rate equal to the speckle coherence time. Therefore each unit must finish its functions within a frame interval. If any unit fails to complete its function in time, the PC is interrupted with an error message.

We plan to integrate for several minutes, typically five minutes, before intermediate results from the averaged flat-fielded image, the 2D FFT, and the power spectrum will be sent to the PC. The processor is designed so that no interruption occurs in the data flow and calculations while data is being transmitted. In this manner a night's observing will not be spoiled by an intermittent problem. The PC will be used to average the final results from the intermediate calculations.

Figure 1. System Block diagram of the Real Time Speckle Processor.

Hardware Implementation: Commercial Components

The processor has been designed to use only commercial CMOS, HCMOS, and some advance CMOS components. We plan to run at a conservative 5 MHz clock rate. However, all our components can operate at much higher speeds.

We use the IDT 39C10, the CMOS equivalent of the AMD 2910, as our micro-code sequencer. The micro-code resides in 2K deep by 96-bits wide dual-ported CMOS memory for all units. With dual-ported memory, we can load and validate the down-loaded micro-code to each unit easily.

Each unit has eight 4096 32-bit wide buffers of static RAM which is used as intermediate memories for the images. Each unit also uses a set of dual-ported memory as ping-pong buffers for communication with the module succeeding it. We designed the ping-pong buffers to operate transparently for both the sender and the receiver.

Integer operations is handled in the frame-grabber by a CMOS version of the TRW 1010J 16-bit multiplier-accumulator chip. Floating-point operations will be performed by an HCMOS 32-bit processor chip, the LSI Logic L64132 from LSI Corp. The flat-fielder and spectrum averager will have a single floating-point chip while the 2D FFT unit requires three floating-point chips.

The interface and the four separate functional units will be wire-wrapped on multi-bus prototype boards by a computer from netlists compiled on a schematic capture program. Two standard multi-bus card cages will be used to house the processor.

2D Complex FFT

Since the 2D complex FFT represents a calculational bottleneck in our processor, we will describe in some detail how we achieved the required speed. Our unit will compute the 2D FFT by using a column-row decomposition. It processes the 64×64 input array by performing a 64-point radix 2 FFT on the 64 columns and storing the result in a corner turning memory. Then the unit performs the FFTs on the 64 rows.

The hardware is optimized to perform a 64-point complex radix-2 FFT. A 64-point radix-2 FFT requires six computational stages (see for example Rabiner and Gold [17]). We divide the hardware into three sections. Each section will calculate two stages of the algorithm. The three sections operate in a pipe-line fashion where section 1 provides the data for section 2, and section 2 provides the data for section 3. The calculations are staggered in time but for the majority of the calculations all three sections are operating in parallel. Dual-ported memories are arranged as ping-pong buffers to store intermediate results between sections. The sine-cosine coefficients and Butterfly addresses are stored in EPROMs.

The computational engine for each section is a 32-bit HCMOS floating-point processor, the LSI Logic L64132. The L64132 can multiply or add two 32-bit floating point numbers in one cycle and output the result in the next cycle. The cycle time for the processor is 125 nsec but we will control it with a 5 MHz (200 nsec) clock. A single node in a complex Butterfly calculation requires 13 clock cycles to perform. The algorithm will be stored in micro-code and a 12-bit CMOS (IDT39C10) sequencer will be used as the controller. We estimate that the 64×64 2D complex FFT can be performed in about 20 msec with this implementation.

We are also collaborating with Prof. Boriakoff, now at Worcester Polytechnic Institute, in the design and fabrication of a chip set which can implement an efficient algorithm for the FFT. The design of the chip is at an advanced stage and will eventually be fabricated through the MOSIS foundry service.

As described in his paper [18], Boriakoff has formulated the Cooley-Tukey scheme into a particular matrix decomposition which is extremely well suited to computations with systolic arrays. In this implementation, the duty cycle of the hardware is improved over that of standard systolic systems (2/3 versus 1/2 of the hardware operating at any given time). In addition, this design is well suited for a pipe-line configuration.

At present the primary effort has gone into the design of a complex inner-product operator which calculates $C_{out} = C_{in} + A \times W$, where C is the partial output, A is the input, and W the sine-cosine coefficients. The design goal is to install one inner-product operator on a single chip (7.9 x 9.2 mm). Specifically, the chip will contain two ROMs for the mantissa and exponents of the sine-cosine coefficients, an adder, a barrel shifter, a 24-bit multiplier, and a normalization section. A set of programmable shift registers are also needed but they will be designed for a second (smaller) chip.

The chips will be implemented on 2 μ m CMOS double metal technology. Although the chips will operate at 35 nsec, we plan to work at 1 μ sec rate. We estimate that a 64×64 complex FFT can be performed in 4 msec with the systolic architecture. The systolic processor requires no programming, no control lines; just a single clock and a gating signal for data input. The processor will do the calculations continuously as data is fed into it. Because of the modular design of our system, the systolic processor can replace our more complicated FFT hardware unit when ready.

Parallel Operations and Programmability

Speed is attained by doing as many operations in parallel as possible. The micro-code word is divided into separate functional fields within each unit. Often each field operates independently of each other and thus can be controlled in parallel. The four units operate in a parallel fashion with the pipe-line design. In addition, the 2D FFT unit has its own three stage pipe-line for calculations. Thus, we designed the processor to achieve efficient throughput of a 64×64 image through various stages of image manipulation and calculation by designing functional parallelism at all levels.

Programmability is a vital property of our processor. We need to implement and improve algorithms for important functions such as flat-fielding that will evolve with experience with a particular detector array. Also the ability to change the function of our modules without hardware alterations is a great advantage. Programming the units also allows us to debug and check the hardware in a simple fashion. Our aim is flexibility in the control of the hardware to perform the tasks we need to do.

We have found in the past that with a micro-code symbolic compiler (Meta-Assembler), programming at the micro-code level is straight forward. Because of the great degree of parallelism in our hardware, the micro-code programs tend to be short; thus we can store our applications well within the 2K limit of our memory.

Conclusion

We have designed and started construction of a real time speckle processor which can perform a 64×64 2D complex FFT on array speckle images within the coherence time of near IR speckles. The design of the processor reflects a great deal of effort to perform the

specialize tasks of speckle observations and the real-time processing of the data in an efficient manner. Our processor is functionally parallel at many levels and the level of performance the processor can achieve is quite high. We have also built into the processor a great deal of flexibility so that improvements in observation techniques and calibration schemes can easily be implemented without hardware changes.

We expect that the processor will have its first telescope trial within the year.

This project is sponsored by the Director's Discretionary Fund at Goddard Space Flight Center.

References

- [1] B. Jones, "Infrared Array Instruments for the Keck Telescope", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 508, 1987.
- [2] A. Labeyrie 1970, *Astr. & Astrophysics*, **6**, 85.
- [3] K.T. Knox and B.J. Thompson, "Recovery of Images from atmospherically Degraded Short-Exposure Photographs," *Ap.J. (Lett.)*, **193**, L45, 1974.
- [4] D.Y. Gezari, A. Labeyrie, and R.V. Stachnik, *Ap. J. (Letters)*, **173**, L1, 1972.
- [5] G. Baier and G. Weigelt, "Speckle interferometric observations of Pluto and its moon Charon on seven different nights", *Astr. & Astrophysics*, **174**, p. 295, 1987.
- [6] P. Nisenson, C. Papaliolios, M. Karovska, and R. Noyes, "Detection of a Very Bright Close Companion to the LMC Supernova SN1987A", *Ap. J. (Letters)*, **320**, L15, 1987.
- [7] W.P.S. Meikle, S.J. Matcher, and B.L. Morgan, *Nature*, **331**, p. 566, 1987.
- [8] P. Nisenson, R.V. Stachnik, C. Papaliolios, and P. Horowitz "Data recording and processing for speckle image reconstruction", *SPIE Vol. 243 Applications of Speckle Phenomena*, p. 88, 1980.
- [9] B.F. Burke, "The Case of the Transient Companion of Supernova SN1987A", submitted to *Nature*, 1988.
- [10] I.S. McLean, "Results with the UKIRT Infrared Camera", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 180, 1987.
- [11] A.M. Fowler, F.C. Gillett, B. Gregory, R.R. Joyce, R.G. Probst, and R. Smith, "The NOAO infrared imagers: Description and performance", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 197, 1987.
- [12] W.J. Forrest, "Ground based imaging with a 32×32 InSb array", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 171, 1987.
- [13] J.C. Christou, J.M. Beckers, F. Roddier, S. Ridgeway, R. Probst, D.W. Freeman, D.W. McCarthy, Jr., and M.L. Cobb, "High Angular resolution infrared imaging at NOAO", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 464, 1987.
- [14] G. Chin and D.Y. Gezari 1987, "High throughput bit-slice front-end processor for the new Goddard 10 μ m 58×62 DRO array camera", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 160.
- [15] A.W. Hoffman, "Operation and calibration of self-integrating multiplexed arrays", Proceedings of the Workshop on Groundbased Astronomical Observations with Infrared Array Detectors, ed. Wynn-Williams and Becklin, p. 29, 1987.
- [16] M.J. McCaughrean, "The Astronomical Applications of Infrared Array Detectors", thesis submitted for the Ph.D. at the University of Edinburgh, UK, 1988.
- [17] L. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing", New York: Prentice Hall, 1975.
- [18] V. Boriakoff, "FFT Computation with Systolic Arrays, A New Architecture", submitted for publication IEEE Proceedings on Computer, 1988.