Adaptive Optics Calculations Using the Connection Machine™

Roger M. Firestone and Eric N. Opp
MRJ, Inc.
10455 White Granite Drive
Oakton, VA 22124

Mark Cullen
Hughes Danbury Optical Systems, Inc.

Abstract

The performance of reflecting optical telescopes located on the surface of the earth are subject to distortions due to the force of gravity on the mirror and the turbulence of the atmosphere on the light path. Reflective optics are also planned for use in high-powered laser systems, where the intensity of the light itself is capable of producing distortions in the air within the instrument, thereby affecting the shape of the focused wavefront. A solution proposed by optical designers is the use of adaptive optics: an optical system in which the figure of the mirror is deformable to the extent necessary to correct for the distortions mentioned. An adaptive optical system uses a feed-back loop concept, in which the distortions of the optical wavefront are measured, the necessary corrections are computed, and a set of actuators is moved to provide those corrections. The calculation of the corrections is computationally intense. Specifically, the measurement of the distortions provides a collection of phase differences between measuring points corresponding to the actuator positions. This set of phase differences is larger than the number of actuators, leading to an overdetermined problem. As physical systems have some amount of noise present, the technique of least-squares solution serves both to provide the best choice of actuator positions for this overdetermined problem and to suppress the noise in the measurements. The necessary algorithms for solving the computation portion of the adaptive optics problem consist of a matrix generator to derive the computational representation of the physical system, a matrix inversion routine, and a high-speed least-squares solver. In the optical astronomy paradigm, the computational requirement is for a small number of adjustments per second, due to the rate of atmospheric turbulence. For the laser system, with more stringent requirements, we demonstrate an improvement of $1\frac{1}{2}$ orders of magnitude, made possible only through the use of supercomputer methods. Extrapolation of these results indicates that even greater acceleration is possible if the interprocessor communication is minimized; in other words, supercomputer designers have not yet solved the problem of making interprocessor communication as efficient as that within processors (or, in the present case, between processors on a single chip).

Introduction

The need to increase the light-gathering power of optical telescopes in order to attain more advanced research objectives mandates ever-larger objective mirrors. Such mirrors encounter two major physical limitations: the force of gravity, which distorts the physical shape of the mirror, and the atmosphere, whose turbulence distorts the shape of the light wavefront before it reaches the mirror. (See Figure 1.) Although neither of these problems affects space-based telescopes, the majority of instruments will be located on the earth's surface for the foreseeable future. The development of adaptive optics addresses both of these difficulties.[1,2]

A third circumstance where adaptive optics has been employed is in the development of high-intensity laser systems. In this design, the light passes through the optics in the opposite direction from that of a telescope. Here, the problem is the distortion due to turbulence caused by the heating produced within the system by the high power of the laser, rather than effects outside the system (which cannot be controlled as their consequences are unknown, taking place after the light leaves the system). The correction process is the same, except that the time scale of the disturbances is considerably shorter than that for gravitational or atmospheric compensation. This is because the size of the disturbances is limited to the size of the system and the rate of fluctuation is correlated with the fluctuation size. Since it is the high-intensity laser systems that impose the most challenging constraints, the present study addresses the computational problems associated with them.

The Laser System

In the laser system there is assumed to be a mirror which may be deformed by a rectangular array of actuators. Corresponding to each actuator is a measuring position. The system determines the optical phase difference between any two adjacent

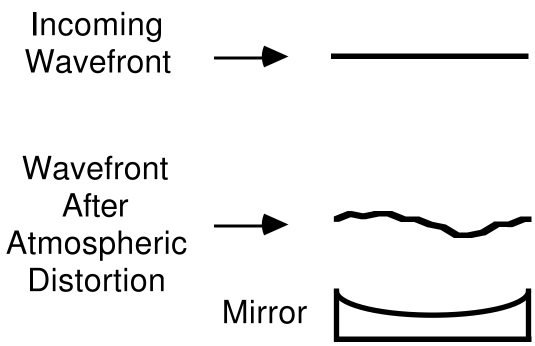


Figure 1. Atmospheric Effect on a Telescope

measuring positions. The set of phase differences represents the departure of the system from ideal optical performance and indicates how the actuators are to be moved to achieve the desired level of performance. In this square lattice structure, there will be slightly less than twice as many phase difference measurements as there are actual phase points; the reconstruction of the phase pattern is thus an overdetermined problem. As noise will be present in the measurements of the phase differences, a least-squares approach is the statistically sound way to solve this overdetermined problem and minimize the effects of the noise.[3]

There are a number of ways to solve least-squares problems, usually based on the normal equations. Typically, these equations are ill-conditioned, requiring solution techniques, such as the QR algorithm, which are slower than the usual method of Gaussian elimination. The adaptive optics problem fortunately does not suffer from this defect. It also has the advantage of a constant solution matrix for a given system hardware design; it is only the data that change. Therefore, the matrix only needs to be inverted once for the entire life of the system (unless changes are made to the complete optical design).

The adaptive optics problem thus has three parts: generation of the constant matrix describing the least-squares problem, inversion of the matrix from the normal equations and formation of its product with the adjoint of the system matrix, and computation of the solution of the least-squares problem. Only the third of these is time-critical.

The Connection Machine

The Connection Machine™ (CM) is a massively-parallel computer manufactured by Thinking Machines Corp. and hosted by a workstation or superminicomputer. A general description of the system may be found in the literature.[4] The present program was developed in a lower-level language (PARIS) for the time-critical portions of the system and in *LISP for the remainder of the system.

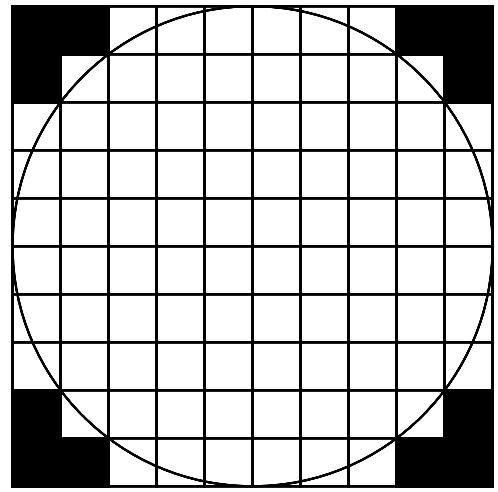


Figure 2. Phase Measurement and
Actuator Array: 10x10 - 12 corner units

Due to the time constraints imposed on the system in the computation of the least-squares solution, it was necessary to make use of low-level hardware features of the Connection Machine not generally described in the literature. Specifically, this meant the use of instructions depending on high-speed communication paths among the processors in the machine. The CM is conceptually organized as a hypercube and is so represented to the *LISP and PARIS programmer; however, its physical organization is that of a hypercube of chips, each chip containing 16 processors. The communication among processors on a single chip ("on-chip" paths) is considerably faster than that between an arbitrary processor pair. There is also a speed advantage if one can limit communications between chips to processors that are directly connected to one another ("single-wire" paths), rather than allowing general communications paths that may involve more than one connection. Both of these techniques were used in this project.

Matrix Generator

The matrix generator is designed to take a description of an optical system and produce a (nonsquare) matrix describing the phase measurement points and relationships in that system. In the present study, a circular mirror is overlaid with a 16x16 grid of actuator/measurement points. This is considerably larger than current adaptive optical systems for observational astronomy and stems from the more demanding nature of the laser system application. Since the mirror is to be circular, fewer than 16^2 = 256 points will actually be present; three points are omitted from each corner of the grid, giving an effective number of 16*16 - 12 = 244 points. The total number of phase gradient measurements is thus 2 * (244-16) = 472, since there are 16 elements that have no neighbor in the vertical direction and 16 elements that have no neighbor in the horizontal direction. Three additional measurements are added, representing the overall tilt of the mirror in the x- and y-directions and the average phase over the entire mirror. This leads to a rectangular matrix of dimension 475x244, which we designate **A**. The least-squares problem may then be expressed as the linear system **Ax** = **y**, where **y** represents the set of phase differences, tilts, and average phase, and **x** is the unknown set of true phases. It is the job of the matrix generator to produce the matrix **A**.

In the present work, it was decided to construct an *ad hoc* matrix generator to handle only the two cases of a 4x4 grid for testing purposes and the full 16x16 matrix for the complete system design. It would be relatively simple to develop a generalized routine to produce a matrix for any size optical system using the ability of the CM to select a subset of processors: Only those processors corresponding to positions of actuators within the dimensions of the mirror would be activated to participate in the development of the matrix. The diagrams show the steps in the construction of the matrix generator for the 4x4 case.

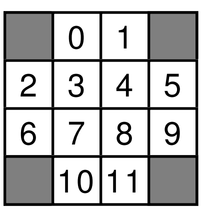
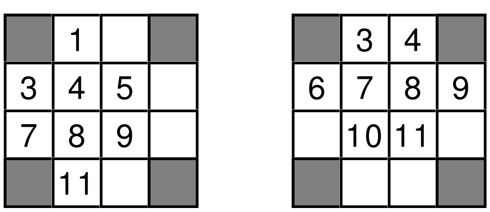


Figure 3. Basic Numbering Scheme



East Neighbors South Neighbors

Figure 4. Neighbor Schemata

It turns out to be computationally more convenient to generate the transpose of the desired matrix, that is, **A***. The matrix is generated by enumerating the active processors—an operation in which each is given a distinct consecutive integer. Neighbors are identified in the horizontal and vertical direction, and each such pair generates a row of the matrix. These rows are all zeros except for a +1 and a -1 in columns identifying the cell and its neighbor, respectively. These entries reflect the measurement of phase difference between neighbors. The tilt and average phase rows are generated by a separate portion of the routine which calculates weights for row, column, and overall averaging.

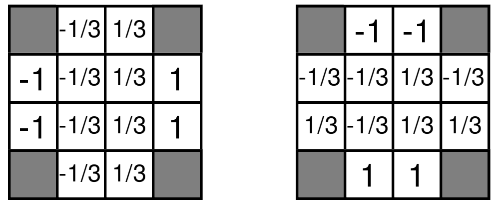


Figure 5. X-tilt (left) and Y-tilt Measurement Weights

Finally, the dot products of the rows of **A*** with themselves in all combinations are calculated in parallel, at each step producing a wrapped diagonal of the product matrix **B** = **A* A**, which represents the normal equations for the least-squares problem. The matrix **B** is square (of dimension 244x244 in the particular case investigated) and real symmetric and is invertible. As noted, the physics of the problem indicate that **B** is not too ill-conditioned to be invertible by conventional methods.

Matrix Inversion

A standard matrix inversion routine was created by another group's research activity. This routine was designed for speed and uses a number of specialized techniques, including the on-chip and single-wire communications methods, as well as conjoined movements of data and indices. The details of this routine are beyond the scope of this paper. The output of this routine is the matrix inverse $\mathbf{B}^{-1}$. At this stage of the program, the product matrix $\mathbf{B}^{-1}\mathbf{A}^*$ is formed, which represents the solution to the least-squares problem for the particular system design.

Least-Squares Calculation

The least-squares calculation consists simply of accepting a vector **y** of phase differences and calculating the phase solution $\mathbf{x} = \mathbf{B}^{-1}\mathbf{A}^*\mathbf{y}$. This seems relatively straightforward; however, the 475*244 = 115900 elements of the matrix $\mathbf{B}^{-1}\mathbf{A}^*$ are more than will fit into even the largest Connection Machine configuration on a one-element per processor basis.

Therefore, the matrix-vector product cannot be formed on a fully parallel basis.

Instead, given the 16384-processor configuration of the CM available, the elements of the matrix were stored eight per processor, making it necessary to perform eight of the multiply-add steps of the matrix-vector multiplication sequentially. The remaining calculations were done in parallel, first accumulating values among the 16 processors within a chip and then accumulating results among processors connected by a single wire.

A detailed study of this algorithm indicates that on the 16K CM there are 8 multiplies, 13 adds, and 6 one-wire communication steps (data movements between processors on the same chip run at the same speed as operand fetch from local processor memory and are not counted). The 65K CM would require 2 multiplies, 9 adds, and 8 one-wire communication steps, reducing the arithmetic operation count by roughly half while increasing the communications time by one-third. Detailed timing studies indicate that the 16K version requires 1351 µseconds on the average, while the 65K version requires a mean time of 1836 µseconds. Thus, **the communications time dominates the arithmetic time**, even though the most efficient single-wire paths were used.

It may be hypothesized that eliminating all inter-chip communications from the algorithm and performing additional arithmetic operations within the chip (32 multiplies, 31 adds, no communications) would provide an additional speed advantage: Treating the computed times as a linear system of equations, we derive a mere 2 µsec per arithmetic operation (and 218 µsec per communication step), which demonstrates the power of the hardware floating-point accelerators available in the later models of the CM. Thus, 63 operations and no communications should be able to reduce the overall timing to well under 1 millisecond, although not as low as 2*63 = 126 µsec due to the presence of other instructions besides arithmetic. Funding limitations, however, did not permit the testing of this version of the algorithm. Figure 6 illustrates these results and extrapolations.

Conclusions

The implementation described in this paper demonstrates that the least-squares problem associated with an adaptive optics system can be solved in as little as 1.351 milliseconds on a 16K Connection Machine, providing a maximum cycle rate of 740 Hertz. This is more than an order of magnitude better than the adjustment rate used in research telescopes and represents a significant advance in the state of the art for adaptive optics. However, the specific application for which this study was undertaken was projected to require a cycle rate in excess of 1 kHz and perhaps as much as 5 kHz. An on-chip algorithm should be able to meet the 1 kHz system requirements and come close to the desired 5 kHz. The role of processor intercommunication delays in implementing fast algorithms continues to be significant.

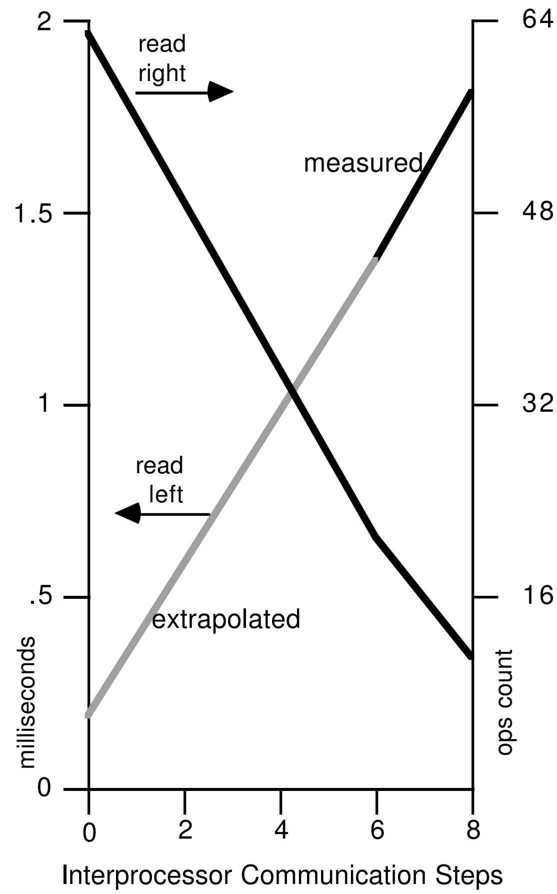


Figure 6. Measured and Extrapolated Performance of Adaptive Optics Algorithm

References

1. Babcock, Horace W. "Adaptive Optics Revisited." *Science* 249:253-7 (20 July 1990).

2. Parenti, Ronald R. "Recent Advances in Adaptive Optics Methods and Technology." *SPIE v. 1000, Laser Wavefront Control.* (1988)

3. Fried, David L. "Least-square Fitting a Wave-front Distortion Estimate…" *J. Opt. Soc. Am.*, vol. 67, no. 3. (1977 March)

4. Hillis, W. Daniel. *The Connection Machine.* M.I.T. Press, Cambridge, MA, 1985.